IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE

POLAROID CORPORATION Plaintiff, v. HEWLETT-PACKARD COMPANY, Defendant.	)	C.A. No. 06-738 (SLR) **REDACTED VERSION**

APPENDIX TO POLAROID CORPORATION'S OPENING BRIEF IN SUPPORT OF ITS MOTION FOR SUMMARY JUDGMENT THAT CLAIMS 1-3 OF U.S. PATENT NO. 4,829,381 ARE NOT OBVIOUS

MORRIS, NICHOLS, ARSHT & TUNNELL LLP
Jack B. Blumenfeld (#1014)
Julia Heaney (#3052)
1201 N. Market Street
Wilmington, Delaware 19801
(302) 658-9200
jheaney@mnat.com

Attorneys for Plaintiff
Polaroid Corporation

OF COUNSEL:

Russell E. Levine, P.C.
G. Courtney Holohan
Michelle W. Skinner
David W. Higer
Maria A. Meginnes
KIRKLAND & ELLIS LLP
200 East Randolph Drive
Chicago, IL 60601
(312) 861-2000

Original Filing Date: May 16, 2008
Redacted Filing Date: May 23, 2008
2331349

INDEX

Ex. A	R.C. Gonzalez and P. Wintz, DIGITAL IMAGE PROCESSING (Addison-Wesley 1987)
Ex. B	U.S. Patent No. 4,654,710 to Christian J. Richard
Ex. C	Jong-Sen Lee, DIGITAL IMAGE ENHANCEMENT AND NOISE FILTERING BY USE OF LOCAL STATISTICS (IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-2, No. 2, pp. 162–68, March 1980)
Ex. D	U.S. Patent No. 4,528,584 to Mohammed S. Sabri
Ex. E	R. Gordon and R.M. Rangayyan , FEATURE ENHANCEMENT OF FILM MAMMOGRAMS USING FIXED AND ADAPTIVE NEIGHBORHOODS (Applied Optics, 1984, 23(4): 560–64)
Ex. F	U.S. Patent No. 4,789,933 to Victor C. Chen and Mike M. Tesic
Ex. G	P.M. Narendra and R.C. Fitch , REAL-TIME ADAPTIVE CONTRAST ENHANCEMENT (IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. PAMI-3, No. 6, pp. 655–61, Nov. 1981)
Ex. H	Appendix A to R.C. Gonzalez and P. Wintz, DIGITAL IMAGE PROCESSING (Addison-Wesley 1987) ("Gonzalez Algorithm")
Ex. I	R. Rangayyan Deposition Transcript
Ex. J	Neil Sullivan *et. al*, A TUTORIAL ON DEVELOPING A COMPUTER-CONTROLLED CAMERA SYSTEM, COMPUTER GRAPHICS, 7–16 (Feb. 1986) (POL 16529–38)
Ex. K	"Real Life Technologies in HP Photosmart Printers and All-in-Ones" (POL 7539015–16)
Ex. L	Stacey Savage *et. al*, "Proposal for Extending Real Life Technologies to Printers" (Dec. 2004) (HPPOL_0266121)
Ex. M	U.S. Patent No. 6,813,041 to Nathan Moroney and Irwin Sobel (POL 7238652–64)

TAB A

DIGITAL IMAGE PROCESSING

Second Edition

Rafael C. Gonzalez
Electrical Engineering Department
University of Tennessee
Knoxville
and
Perceptics Corporation
Knoxville, Tennessee

Paul Wintz
Consultant

ADDISON-WESLEY PUBLISHING COMPANY

Reading, Massachusetts · Menlo Park, California
Don Mills, Ontario · Wokingham, England
Amsterdam · Sydney · Singapore · Tokyo
Madrid · Bogotá · Santiago · San Juan

Library of Congress Cataloging-in-Publication Data

Gonzalez, Rafael C.
Digital image processing.

Bibliography: p.
Includes index.
1. Image processing—Digital techniques. I. Wintz, Paul A. II. Title.
TA1632.G66 1987 621.36'7 86-28759
ISBN 0-201-11026-1

Reprinted with corrections November, 1987

HIJ-MA-89

CHAPTER 1

INTRODUCTION

One picture is worth more than ten thousand words.
Anonymous

1.1 BACKGROUND

Interest in digital image processing methods stems from two principal application areas: improvement of pictorial information for human interpretation, and processing of scene data for autonomous machine perception. One of the first applications of image processing techniques in the first category was in improving digitized newspaper pictures sent by submarine cable between London and New York. Introduction of the Bartlane cable picture transmission system in the early 1920s reduced the time required to transport a picture across the Atlantic from more than a week to less than three hours. Pictures were coded for cable transmission and then reconstructed at the receiving end by specialized printing equipment. Figure 1.1 was transmitted in this way and reproduced on a telegraph printer fitted with type faces simulating a halftone pattern.

Some of the initial problems in improving the visual quality of these early digital pictures were related to the selection of printing procedures and the distribution of brightness levels. The printing method used to obtain Fig. 1.1 was abandoned toward the end of 1921 in favor of a technique based on photographic reproduction made from tapes perforated at the telegraph receiving terminal. Figure 1.2 shows an image obtained using this method. The improvements over Fig. 1.1 are evident, both in tonal quality and in resolution.

The early Bartlane systems were capable of coding images in 5 distinct brightness levels. This capability was increased to 15 levels in 1929. Figure 1.3 is indicative of the type of image that could be obtained using the 15-tone equipment. During this period, the reproduction process was also improved considerably by the introduc-

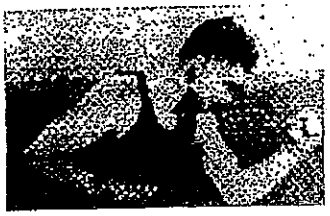


Figure 1.1 A digital picture produced in 1921 from a coded tape by a telegraph printer with special type faces. (From McFarlane [1972].)

tion of a system for developing a film plate via light beams that were modulated by the coded picture tape.

Although improvements on processing methods for transmitted digital pictures continued to be made over the next 35 years, it took the combined advents of large-scale digital computers and the space program to bring into focus the potentials of image processing concepts. Work on using computer techniques for improving images from a space probe began at the Jet Propulsion Laboratory (Pasadena, Calif.) in 1964, when pictures of the moon transmitted by Ranger 7 were processed by a computer to correct various types of image distortion inherent in the on-board television camera. These techniques served as the basis for improved methods used in the enhancement and restoration of images from such familiar programs as the Surveyor missions to the moon, the Mariner series of flyby missions to Mars, and the Apollo manned flights to the moon.

From 1964 until this writing, the field of image processing has experienced vigorous growth. In addition to applications in the space program, digital image processing techniques are used today in a variety of problems that, although often unrelated, share a common need for methods capable of enhancing pictorial information for human interpretation and analysis. In medicine, for instance, physicians are assisted by computer procedures that enhance the contrast or code the intensity levels into color for easier interpretation of x-rays and other biomedical images. The same or similar techniques are used by geographers in studying pollution patterns from aerial and satellite imagery. Image enhancement and restoration procedures

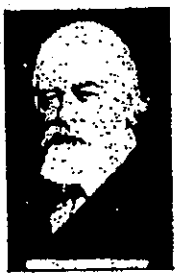


Figure 1.2 A digital picture made from a tape punched after the signals had crossed the Atlantic twice. Some errors are visible. (From McFarlane [1972].)

Figure 1.3 Unretouched cable picture of Generals Pershing and Foch, transmitted by 15-tone equipment from London to New York. (From McFarlane [1972].)

have been used to process degraded images depicting unrecoverable objects or experimental results too expensive to duplicate. There have been instances in archeology, for example, where blurred pictures that were the only available records of rare artifacts lost or damaged after being photographed, have been successfully restored by image processing methods. In physics and related fields, images of experiments in such areas as high-energy plasmas and electron microscopy are routinely enhanced by computer techniques. Similar successful applications of image processing concepts can be found in astronomy, biology, nuclear medicine, law enforcement, defense, and industrial applications.

Some typical examples of the results obtainable with digital image processing techniques are shown in Fig. 1.4. The original images are shown on the left and the corresponding computer-processed images on the right. Figure 1.4(a) is a picture of the Martian surface that was corrupted by interference during transmission to Earth by a space probe. The interference, which in this case appears as a set of vertical, structured lines, can be almost completely removed by computer processing, as shown in Fig. 1.4(b). Figures 1.4(c) and (d) illustrate the considerable improvement that can be made on an x-ray image by contrast and edge enhancement. The image shown in Fig. 1.4(e) was blurred by uniform motion during exposure, and the image shown in Fig. 1.4(f) resulted after application of a deblurring algorithm. These illustrations are typical of those discussed in detail in Chapters 4 and 5.

The foregoing examples have in common the fact that processing results are intended for human interpretation. The second major area of application of digital image processing techniques mentioned at the beginning of this section is in problems dealing with machine perception. In this case, interest is focused on procedures for

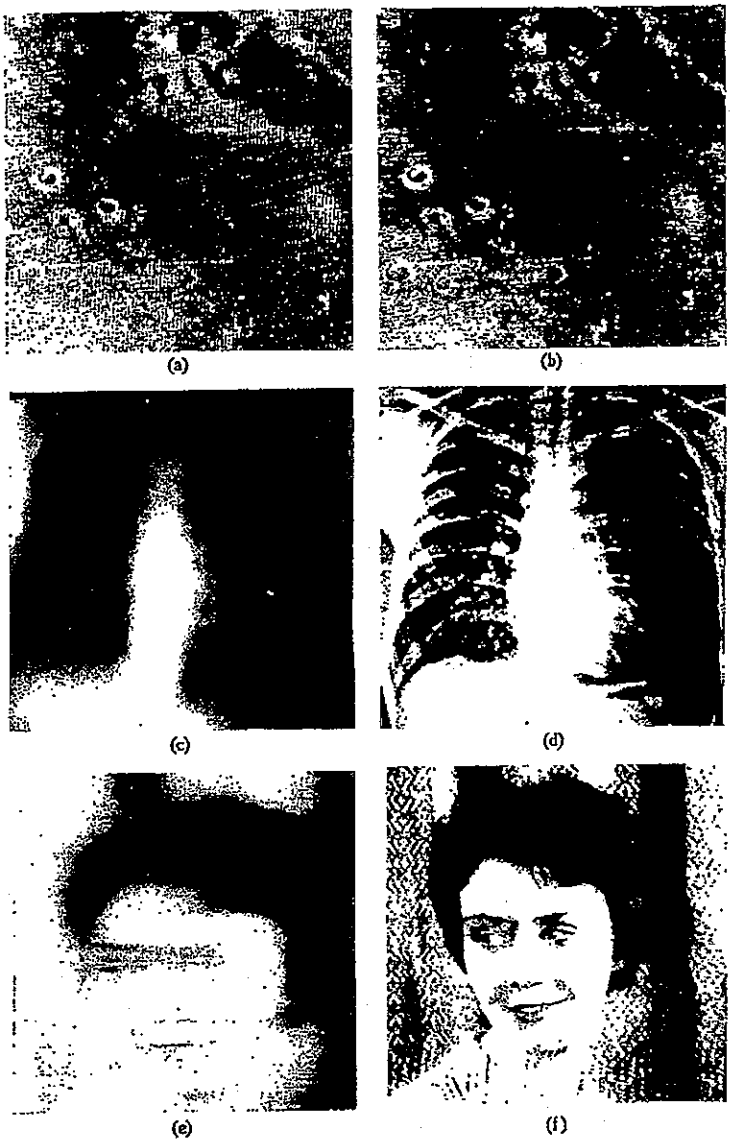


Figure 1.4 Examples of image processing. Left column: original digital images. Right column: images after processing.

extracting from an image information in a form suitable for computer processing. Often, this information bears little resemblance to visual features used by humans in interpreting the content of an image. Examples of the type of information used in machine perception are statistical moments, Fourier transform coefficients, and multidimensional distance measures.

Typical problems in machine perception that routinely employ image processing techniques are automatic character recognition, industrial robots for product assembly and inspection, military recognizance, automatic processing of fingerprints, screening of x-rays and blood samples, and machine processing of aerial and satellite imagery for weather prediction and crop assessment.

1.2 DIGITAL IMAGE REPRESENTATION

As used in this book, the term *monochrome image* or simply *image*, refers to a two-dimensional light intensity function $f(x, y)$, where x and y denote spatial coordinates and the value of f at any point (x, y) is proportional to the brightness (or *gray level*) of the image at that point. An example illustrating the axis convention used throughout the following chapters is shown in Fig. 1.5. It is sometimes useful

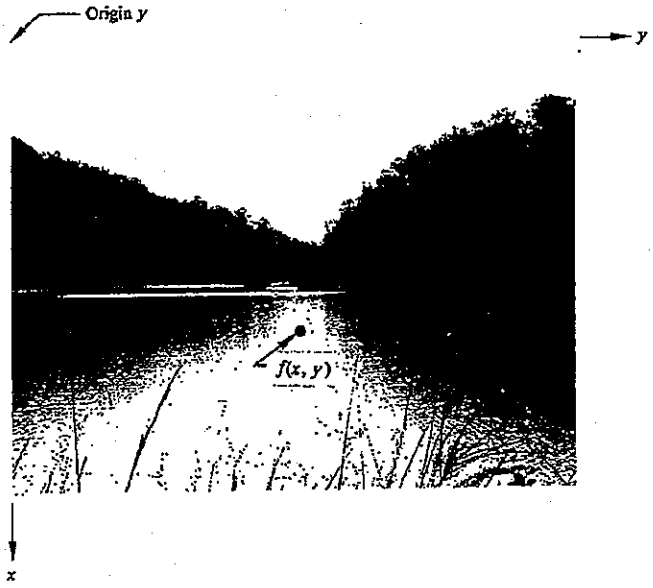


Figure 1.5 Axis convention used for digital image representation.

to view an image function in perspective with the third axis being brightness. If Fig. 1.5 were viewed in this way it would appear as a series of active peaks in regions with numerous changes in brightness levels and smoother regions or plateaus where the brightness levels varied little or were constant. If we follow the convention of assigning proportionately higher values to brighter areas, the height of the components in the plot would be proportional to the corresponding brightness in the image.

A *digital image* is an image $f(x, y)$ that has been discretized both in spatial coordinates and in brightness. We may consider a digital image as a matrix whose row and column indices identify a point in the image and the corresponding matrix element value identifies the gray level at that point. The elements of such a digital array are called *image elements, picture elements, pixels,* or *pels,* with the last two names being commonly used abbreviations of "picture elements."

Although the size of a digital image varies with the application, it will become evident in the following chapters that there are numerous advantages to selecting square arrays with sizes and number of gray levels that are integer powers of 2. For example, a typical size comparable in quality to a monochrome TV image is a 512×512 array with 128 gray levels.

With the exception of a discussion in Chapter 4 of pseudo-color techniques for image enhancement, all the images considered in this book are digital monochrome images of the form described above. Thus we will not be concerned with topics in three-dimensional scene analysis nor with optical techniques for image processing.

1.3 ELEMENTS OF A DIGITAL IMAGE PROCESSING SYSTEM

The components of a basic, general-purpose digital image processing system are shown in Fig. 1.6. The operation of each block in Fig. 1.6 is explained briefly below.

1.3.1 Image Processors

A digital image processor is the heart of any image processing system. An image processor consists of a set of hardware modules that perform four basic functions: image acquisition, storage, low-level (fast) processing, and display. Typically, the image acquisition module has a TV signal as the input (see Section 1.3.2) and converts this signal into digital form, both spatially and in amplitude (see Section 2.3). Most modern image processors are capable of digitizing a TV image in one frame-time (i.e., 1/30th of a second). For this reason, the image acquisition module is often referred to as a *frame grabber*.

The storage module, often called a *frame buffer,* is a memory capable of storing an entire digital image. Usually, several such modules are incorporated in an image processor. The single most distinguishing characteristic of an image storage module is that the contents of the memory can be loaded or read at TV rates (on the order of 30 images per second). This feature allows the image acquisition module

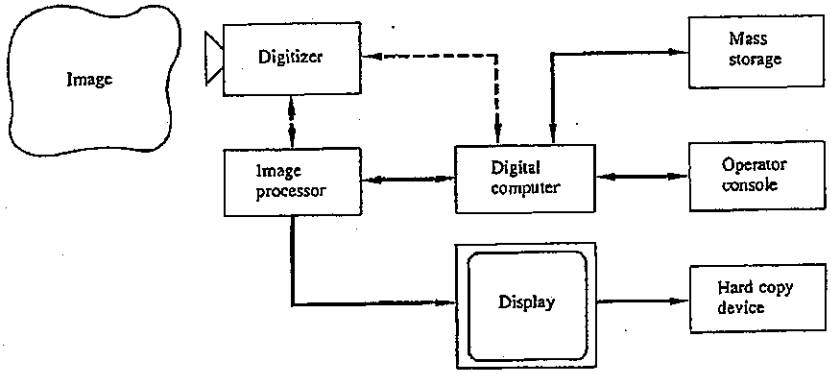


Figure 1.6 Elements of a digital image processing system. The dashed lines indicate that typically only one of the two connections is made.

deposit a complete image into storage as fast as it is being grabbed. Conversely, the memory can be addressed at TV rates by a display module, which outputs the image to a TV monitor, as discussed in Section 1.3.5. Other memory-addressing modes allow virtually instantaneous image *zoom*, as well as *scroll* (vertical shifts) and *pan* (horizontal shifts).

The processing module performs low-level functions such as arithmetic and logic operations. Thus this module is often called an *Arithmetic-Logic Unit* (ALU). It is a specialized hardware device designed specifically to gain speed by processing pixels in parallel. The function of the display module is to read an image memory, convert the stored digital information into an analog video signal, and output this signal to a TV monitor or other video device. Typical additional hardware display options include gray-level transformation functions and graphics, as well as alphanumeric overlays.

1.3.2 Digitizers

A digitizer converts an image into a numerical representation suitable for input into a digital computer. Among the most commonly used input devices are microdensitometers, flying spot scanners, image dissectors, vidicon cameras, and photosensitive solid-state arrays. The first two devices require that the image to be digitized be in the form of a transparency (e.g., a film negative) or photograph. Image dissectors, vidicon cameras, and solid-state arrays can accept images recorded in this manner, but they have the additional advantage of being able to digitize natural images that have sufficient light intensity to excite the detector.

In microdensitometers the transparency or photograph is mounted on a flat bed or wrapped around a drum. Scanning is accomplished by focusing a beam of light on the image and translating the bed or rotating the drum in relation to the beam. In the case of transparencies the beam passes through the film; in photographs it is reflected from the surface of the image. In both cases the beam is focused on a photodetector and the gray level at any point in the image is recorded by the detector based on the intensity of the beam. A digital image is obtained by allowing only discrete values of intensity and position in the output. Although microdensitometers are slow devices, they are capable of high degrees of position accuracy due to the essentially continuous nature of the mechanical translation used in the digitization process.

Flying spot scanners also operate on the principle of focusing a transmitted or reflected source beam on a photodetector. In this case, however, the image is stationary and the light source is a cathode-ray tube (CRT) in which a beam of electrons, deflected by electromagnets, impinges on a fluorescent phosphor surface. The beam thereby produces a spot of light that moves in a scanning pattern on the face of the tube. The fact that the beam is moved electronically allows high scanning speeds. Flying spot scanners are also ideally suited for applications in which it is desirable to control the beam scanning pattern externally (e.g., in tracing the boundaries of objects in an image). This flexibility is afforded by the fact that the position of the electron beam is quickly and easily established by external voltage signals applied to the electromagnets.

In image dissectors and vidicon cameras the image is focused directly on the surface of a photosensitive tube whose response is proportional to the incident light pattern. Dissector operation is based on the principle of electronic emission, where the image incident on the photosensitive surface produces an electron beam whose cross section is roughly the same as the geometry of the tube surface. Image pickup is accomplished by using electromagnets to deflect the entire beam past a pinhole located in the back of the dissector tube. The pinhole lets through only a small cross section of the beam and thus "looks" at one point in the image at a time. Since photoemissive materials are very inefficient, the time that the pinhole has to look at the point source in order to collect enough electrons tends to make image dissectors rather slow digitizers. Most devices integrate the emission of each input point over a specified time interval before yielding a signal that is proportional to the brightness of the point. This integration capability is beneficial in terms of noise reduction, thus making image dissectors attractive in applications where high signal-to-noise ratios are required. As in flying spot scanners, control of the scanning pattern in image dissectors is easily varied by external voltage signals applied to the electromagnets.

The operation of vidicon cameras is based on the principle of photoconductivity. An image focused on the tube surface produces a pattern of varying conductivity that matches the distribution of brightness in the optical image. An independent,

finely focused electron beam scans the rear surface of the photoconductive target, and, by charge neutralization, this beam creates a potential difference that produces on a collector a signal proportional to the input brightness pattern. A digital image is obtained by quantizing this signal, as well as the corresponding position of the scanning beam.

Solid-state arrays are composed of discrete silicon imaging elements, called *photosites*, that have a voltage output proportional to the intensity of the incident light. Solid-state arrays are organized in one of two principal geometrical arrangements: *line scan sensors* and *area sensors*. A line scan sensor consists of a row of photosites and produces a two-dimensional image by relative motion between the scene and the detector. An area sensor is composed of a matrix of photosites and is therefore capable of capturing an image in the same manner as, say, a vidicon tube.

Vidicon and area sensors are typically packaged as TV cameras. Image digitization is achieved by feeding the output of the camera into the image acquisition module, as discussed in the previous section. Although TV cameras are in general less accurate than the systems discussed above, they have numerous advantages that in many applications outweigh their relative lack of precision. Vidicon systems, for example, are among the most inexpensive digitizers in the market. They also have the distinct advantage that the image being digitized can be viewed in its entirety on a TV monitor. This capability, not available in any of the systems discussed above, is ideal for general-purpose applications.

1.3.3 Digital Computers

Although, as mentioned in Section 1.3.1, an image processor may be equipped with internal processing capabilities, the level of this processing is rather low in sophistication. Thus one usually finds that image processors are interfaced to a general-purpose computer, which provides versatility as well as ease of programming. Computer systems used for image processing range from microprocessor devices to large computer systems capable of performing computationally intensive functions on large image arrays. The principal parameters influencing the structure of a computer for image processing are the intended application and the required data throughput. For dedicated applications (which normally dictate low cost) a well-equipped microcomputer or minicomputer will often be sufficient. If the application involves extensive program development or is characterized by significant data throughputs, a mainframe computer would most likely be required. In this case, a computer with virtual memory addressing capabilities has significant advantages. Virtual addressing makes disk peripheral storage available to the user as if it were main memory. This feature, which is transparent to the user, is of crucial importance because digital images utilize large amounts of memory during processing. The alternative to virtual addressing is a set of user-supplied complex routines whose only function is to swap image segments in and out of peripheral storage during processing.

1.3.4 Storage Devices

A digital image consisting of 512 × 512 pixels, each of which is quantized i
eight bits, requires 0.25 megabytes of storage. Thus providing adequate bulk stor
facilities is one of the most important aspects in the design of a general-purp
image processing system. The three principal storage media used in this type
work are magnetic disks, magnetic tapes, and optical disks. Magnetic disks wit
capacity of 700 megabytes or more are common. A 700-megabyte disk would h
on the order of 2800 images of the size mentioned above. High-density magn
tapes (6400 bytes per inch) can store one such image in approximately four feet
tape. Optical disks, which are based on laser writing and reading technology, h
recently become commercially available. The storage capacity of a single opt
disk platter can approach 4 gigabytes, which translates into approximately 16,
images per disk.

1.3.5 Display and Recording Devices

Monochrome and color television monitors are the principal display devices u
in modern image processing systems. Monitors are driven by the output(s) of
image display module in the image processor, as discussed in Section 1.3.1. Th
signals can also be fed into an image recording device whose function is to proc
a hard copy (slides, photographs, and transparencies) of the image being vie
on the monitor screen. Other display media include CRTs and printing devices.

In CRT systems the horizontal and vertical positions of each element in
image array are converted into voltages that are used to deflect the CRT's elec
beam, thus providing the two-dimensional drive necessary to produce an ou
image. At each deflection point, the intensity of the beam is modulated by usi
voltage that is proportional to the value of the corresponding point in the nume
array, varying from zero intensity outputs for points whose numerical value c
sponds to black, to maximum intensity for white points. The resulting vari
intensity light pattern is recorded by a photographic camera focused on the fa
the cathode-ray tube. Some systems employ a long-persistence phosphor tube, w
also allows viewing of the entire image after the scanning process is compl
Although images recorded by a photographic process can be of excellent qua
the same images generally appear of poor tonality when shown to an observer
long-persistence CRT because of limitations in the human visual system when resp
ing to this type of display.

Printing image display devices are useful primarily for low-resolution i
processing work. One simple approach for generating gray-tone images direct
paper is to use the overstrike capability of a standard line printer. The gray
of any point in the printout can be controlled by the number and density c
characters overprinted at that point. By properly selecting the character set
possible to achieve reasonably good gray-level distributions with a simple com
program and relatively few characters. An example of this approach is giv

Appendix A. Other common means of recording an image directly on paper include laser printers, heat-sensitive paper devices, and ink-spray systems.

1.4 ORGANIZATION OF THE BOOK

Techniques for image processing may be divided into four principal categories: (1) image digitization; (2) image enhancement and restoration; (3) image encoding; and (4) image segmentation, representation, and description. The material in the following chapters is organized in essentially the same order as these problem areas.

As discussed in Sections 1.2 and 1.3, the digitization problem is one of converting continuous brightness and spatial coordinates into discrete components. A preliminary discussion of digitization and its effect on image quality is given in Chapter 2, while a more theoretical treatment of the sampling process is developed in Chapter 3. Digitization considerations are a natural extension of the main theme of these two chapters, which is the introduction of concepts and mathematical tools used throughout the rest of the book.

Enhancement and restoration techniques deal with the improvement of a given image for human or machine perception. Image enhancement is the topic of Chapter 4, while image restoration methods are covered in Chapter 5. Image encoding procedures, discussed in Chapter 6, are used to reduce the number of bits in a digital image. The encoding process often plays a central role in image processing for the purpose of minimizing storage or transmission requirements. Segmentation techniques are considered in Chapter 7. Segmentation is the process that subdivides an image into its constituent regions or objects. Chapter 8 deals with representation and description, which are important processes in the implementation of autonomous image processing and analysis systems.

Two appendices are included at the end of the book. Appendix A contains a set of FORTRAN routines for displaying images on an ordinary line printer. Appendix B contains a set of coded digital images, which together with the routines in Appendix A, can be used to test with pictorial data the methods developed in the book.

REFERENCES

The references cited below are of a general nature and cover the spectrum of available image processing techniques and their applications. References given at the end of later chapters are keyed to specific topics discussed in the text. All references are cited by author, book, or journal name followed by the year of publication. The bibliography at the end of the book is organized in the same way and contains all pertinent information for each reference.

Some of the major journals that publish articles on image processing and related topics include: *Computer Vision, Graphics, and Image Processing, IEEE Transactions on Systems, Man and Cybernetics, IEEE Transactions on Pattern Analysis and Machine Intelligence, Pattern Recognition, IEEE Transactions on Medical Imaging, Journal of the Optical Society of America, IEEE Transactions on Information Theory, IEEE Transactions on Communications, IEEE Transac-*

tions on Acoustics, Speech and Signal Processing, Proceedings of the IEEE, and issues of *IEEE Transactions on Computers* prior to 1980.

Other image processing books of interest include Andrews [1970], Pratt [1978], Castlei [1979], Hall [1979], and Rosenfeld and Kak [1982]. The literature on pattern recognition o contains articles related to image processing. The books by Duda and Hart [1973], Tou Gonzalez [1974], Pavlidis [1977], Gonzalez and Thomason [1978], and Fu [1982] contai guide to the literature on pattern recognition and related topics.

CHAPTER 2

DIGITAL IMAGE FUNDAMENTALS

Those who wish to succeed must ask the right preliminary questions.
Aristotle

The purpose of this chapter is to introduce the reader to a number of image concepts and to develop some of the notation that will be used throughout the book. The first section is a brief summary of the mechanics of the human visual system, including image formation in the eye and its capabilities for brightness adaptation and discrimination. Section 2.2 presents an image model based on the illumination–reflection phenomenon, which gives rise to most images perceived in our normal visual activities. The concepts of uniform image sampling and gray-level quantization are introduced in Section 2.3. Section 2.4 deals with relationships between pixels, such as connectivity and distance measures, which will be used extensively throughout the book. Section 2.5 contains a detailed discussion of imaging geometry and related topics. Finally, Section 2.6 contains an introduction to photographic film and some of its most important characteristics in terms of recording image processing results.

2.1 ELEMENTS OF VISUAL PERCEPTION

Since the ultimate goal of many of the techniques discussed in the following chapters is to aid the observer in interpreting the content of an image, it is important before proceeding to develop a basic understanding of the visual-perception process. The following discussion is a brief account of the human visual mechanism, with particular emphasis on concepts that will serve as a foundation to much of the material presented in later chapters.

2.1.1 Structure of the Human Eye

A horizontal cross section of the human eye is shown in Fig. 2.1. The eye nearly spherical in form with an average diameter of approximately 20 mm. I enclosed by three membranes: the *cornea* and *sclera* outer cover, the *choroid*, the *retina*. The cornea is a tough, transparent tissue that covers the anterior surf of the eye. The sclera is continuous with the cornea; it is an opaque membr that encloses the remainder of the optic globe.

The choroid lies directly below the sclera. This membrane contains a netw of blood vessels that serve as the major source of nutrition to the eye. The cho coat is heavily pigmented and hence helps to reduce the amount of extrane

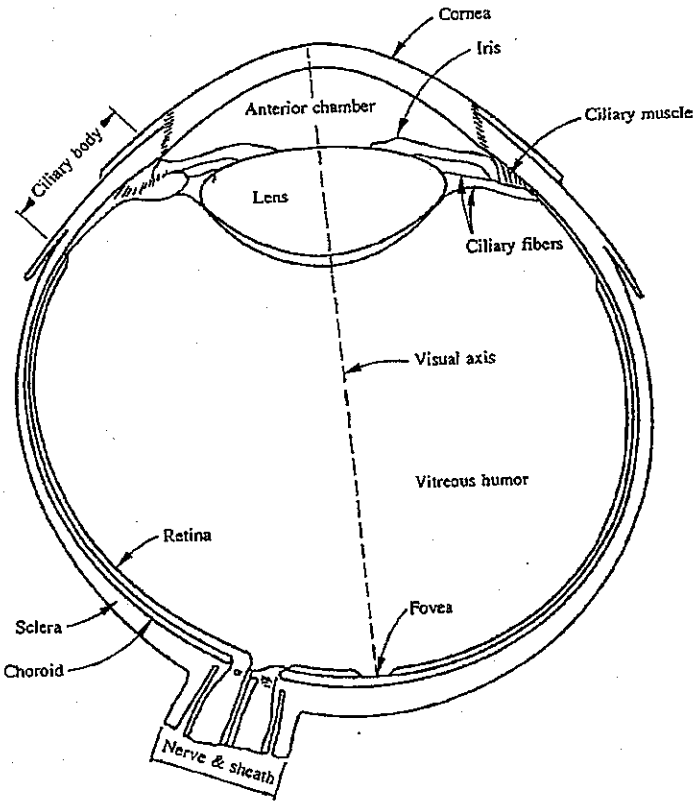


Figure 2.1 Simplified diagram of a cross section of the human eye.

light entering the eye and the backscatter within the optical globe. At its anterior extreme, the choroid is divided into the *ciliary body* and the *iris diaphragm*. The latter contracts or expands to control the amount of light that is permitted to enter the eye. The central opening of the iris (the *pupil*) is variable in diameter from approximately 2 mm up to 8 mm. The front of the iris contains the visible pigment of the eye, whereas the back contains a black pigment.

The innermost membrane of the eye is the retina, which lines the inside of the wall's entire posterior portion. When the eye is properly focused, light from an object outside the eye is imaged on the retina. Pattern vision is afforded by the distribution of discrete light receptors over the surface of the retina. There are two classes of receptors: *cones* and *rods*. The cones in each eye number between 6 and 7 million. They are located primarily in the central portion of the retina, called the *fovea*, and are highly sensitive to color. Humans can resolve fine details with these cones largely because each one is connected to its own nerve end. Muscles controlling the eye rotate the eyeball until the image of an object of interest falls on the fovea. Cone vision is known as *photopic* or bright-light vision.

The number of rods is much larger, being on the order of 75 to 150 million distributed over the retinal surface. The larger area of distribution and the fact that several rods are connected to a single nerve end reduce the amount of detail discernible by these receptors. Rods serve to give a general, overall picture of the field of view. They are not involved in color vision and are sensitive to low levels of illumination. For example, objects that appear brightly colored in daylight, when seen by moonlight appear as colorless forms because only the rods are stimulated. This is known as *scotopic* or dim-light vision.

The *lens* is made up of concentric layers of fibrous cells and is suspended by fibers that attach to the ciliary body. It contains 60 to 70 percent water, about 6 percent fat, and more protein than any other tissue in the eye. The lens is colored by a slightly yellow pigmentation that increases with age. It absorbs approximately 8 percent of the visible light spectrum, with relatively higher absorption at shorter wavelengths. Both infrared and ultraviolet light are absorbed appreciably by proteins within the lens structure and, in excessive amounts, can cause damage to the eye.

2.1.2 Image Formation in the Eye

The principal difference between the lens of the eye and an ordinary optical lens is that the former is flexible. As illustrated in Fig. 2.1, the radius of curvature of the anterior surface of the lens is greater than the radius of its posterior surface. The shape of the lens is controlled by the tension in the fibers of the ciliary body. To focus on distant objects, the controlling muscles cause the lens to be relatively flattened. Similarly, these muscles allow the lens to become thicker in order to focus on objects near the eye.

The distance between the focal center of the lens and the retina varies from approximately 17 mm down to about 14 mm, as the refractive power of the lens increases from its minimum to its maximum. When the eye is focused on an object

farther than about 3 m away, the lens exhibits its lowest refractive power, a when focused on a very near object it is most strongly refractive. With this informati it is easy to calculate the size of the retinal image of any object. In Fig. 2.2, example, the observer is looking at a tree 15 m high at a distance of 100 Letting x be the size of the retinal image in millimeters, we have from the geome of Fig. 2.2 that $15/100 = x/17$ or $x = 2.55$ mm. As indicated in the previ section, the retinal image is reflected primarily in the area of the fovea. Percept then takes place by the relative excitation of light receptors, which transform radi energy into electrical impulses that are ultimately decoded by the brain.

2.1.3 Brightness Adaptation and Discrimination

Since digital images are displayed as a discrete set of brightness points, the abi of the eye to discriminate between different brightness levels is an important consi ation in presenting image processing results.

The range of light intensity levels to which the human visual system can ac is enormous, being on the order of 10^{10} from the scotopic threshold to the g limit. There is also considerable experimental evidence that indicates that subjec brightness (i.e., brightness as perceived by the human visual system) is a logarith function of the light intensity incident on the eye. This characteristic is illustr in Fig. 2.3, which is a plot of light intensity versus subjective brightness. long solid curve represents the range of intensities to which the visual system adapt. In photopic vision alone, the range is about 10^6. The transition from scot to photopic vision is gradual over the approximate range from 0.001 to 0.1 millilam (−3 to −1 mL in the log scale), as illustrated by the double branches of the adapta curve in this range.

The key point in interpreting the impressive dynamic range depicted in 2.3 is that the visual system can by no means operate over such a range *simultaneo* Rather, it accomplishes this large variation by changes in its overall sensitivit

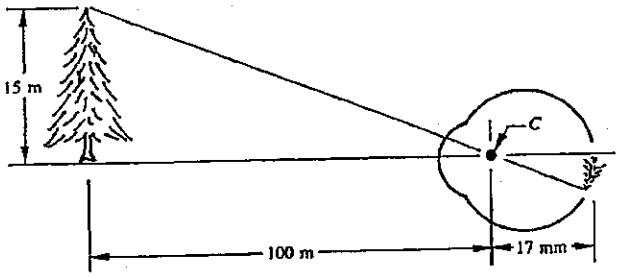


Figure 2.2 Optical representation of the eye looking at a tree. Point C is the optical center of the lens.

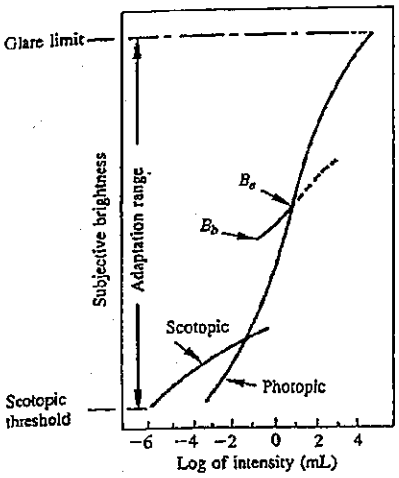


Figure 2.3 Range of subjective brightness sensations showing a particular adaptation level.

phenomenon known as *brightness adaptation*. The total range of intensity levels it can discriminate simultaneously is rather small when compared with the total adaptation range. For any given set of conditions, the current sensitivity level of the visual system is called the *brightness-adaptation level*, which may correspond, for example, to brightness B_a in Fig. 2.3. The short intersecting curve represents the range of subjective brightness that the eye can perceive when adapted to this level. It is noted that this range is rather restricted, having a level B_b at and below which all stimuli are perceived as indistinguishable blacks. The upper (dashed) portion of the curve is not actually restricted but, if extended too far, loses its meaning because much higher intensities would simply raise the adaptation level to a higher value than B_a.

The contrast sensitivity of the eye can be measured by exposing an observer to a uniform field of light of brightness B, with a sharp-edged circular target in the center, of brightness $B + \Delta B$, as shown in Fig. 2.4(a). ΔB is increased from zero until it is just noticeable. The just-noticeable difference ΔB is measured as a function of B. The quantity $\Delta B/B$ is called the Weber ratio and is nearly constant at about 2 percent over a very wide range of brightness levels, as shown in Fig. 2.4(b). This phenomenon has given rise to the idea that the human eye has a much wider dynamic range than manmade imaging systems. However, this does not correspond to any ordinary seeing situation and more-applicable results are obtained by using the pattern of Fig. 2.5(a). $\Delta B/B$ is again measured, but now B_0, the surrounding (adapting) brightness, is a parameter. The results are shown in Fig. 2.5(b). The dynamic range is about 2.2 log units centered about the adapting brightness, which is compara-

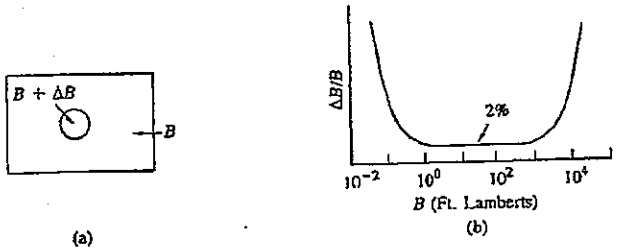


Figure 2.4 Contrast sensitivity with a constant background.

ble to what can be achieved with electronic imaging systems if they are corre adjusted for the background brightness. The ease and rapidity with which the adapts itself—differently on different parts of the retina—is really the remarka characteristic, rather than its overall dynamic range. What is meant by a dyna range of 2.2 log units is that $\Delta B/B$ remains relatively constant in this range. A becomes more and more different from the adapting brightness B_0, the appeara also changes. Thus a brightness about 1.5 log units higher or lower than B_0 app white or black, respectively. If the central target is set at a constant level whil is varied over a wide range, the target appears to change from completely whit completely black.

In the case of a complex image, the visual system does not adapt to a si intensity level. Instead, it adapts to an average level that depends on the prope of the image. As the eye roams about the scene, the instantaneous adaptation l fluctuates about this average. For any point or small area in the image, the W ratio is generally much larger than that obtained in an experimental environ because of the lack of sharply defined boundaries and intensity variations ir

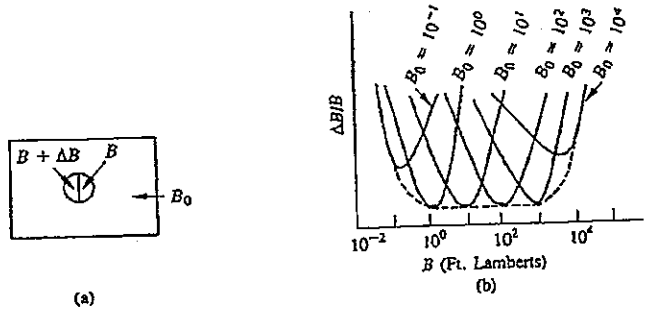


Figure 2.5 Contrast sensitivity with a varying background.

background. The result is that the eye can detect only in the neighborhood of one or two dozen intensity levels at any one point in a complex image. This does not mean, however, that an image need only be displayed in two dozen intensity levels to achieve satisfactory visual results. The above narrow discrimination range "tracks" the adaptation level as the latter changes in order to accommodate different intensity levels following eye movements about the scene. This allows a much larger range of *overall* intensity discrimination. To obtain displays that will appear reasonably smooth to the eye for a large class of image types, a range of over 100 intensity levels is generally required. This point will be considered in further detail in Section 2.3.

The brightness of a region, as perceived by the eye, depends on factors other than simply the light radiating from that region. In terms of image processing applications, one of the most interesting phenomena related to brightness perception is that the response of the human visual system tends to "overshoot" around the boundary of regions of different intensity. The result of this overshoot is to make areas of constant intensity appear as if they had varying brightness. In Fig. 2.6(a), for example, the image shown was created by varying the intensity according to the intensity profile shown below the photograph. Although the intensity variation is perfectly smooth, the eye perceives a brighter stripe in the region marked B and a darker stripe in the region marked D. These stripes are called *Mach bands*, after Ernst Mach who first described them in 1865. A more striking example of the Mach-band effect is shown in Fig. 2.6(b). As indicated by the intensity profile, each band in the photograph was created by using a constant intensity. To the eye, however, the brightness pattern in the image appears strongly scalloped, particularly around the boundaries.

2.2 AN IMAGE MODEL

As used in this book, the term *image* refers to a two-dimensional light-intensity function, denoted by $f(x, y)$, where the value or amplitude of f at spatial coordinates (x, y) gives the intensity (brightness) of the image at that point. Since light is a form of energy, $f(x, y)$ must be nonzero and finite, that is,

$$0 < f(x, y) < \infty. \tag{2.2-1}$$

The images we perceive in our everyday visual activities normally consist of light reflected from objects. The basic nature of $f(x, y)$ may be considered as being characterized by two components. One component is the amount of source light incident on the scene being viewed, while the other is the amount of light reflected by the objects in the scene. These components are appropriately called the *illumination* and *reflectance components*, and are denoted by $i(x, y)$ and $r(x, y)$, respectively. The functions $i(x, y)$ and $r(x, y)$ combine as a product to form $f(x, y)$:

$$f(x, y) = i(x, y)r(x, y), \tag{2.2-2}$$

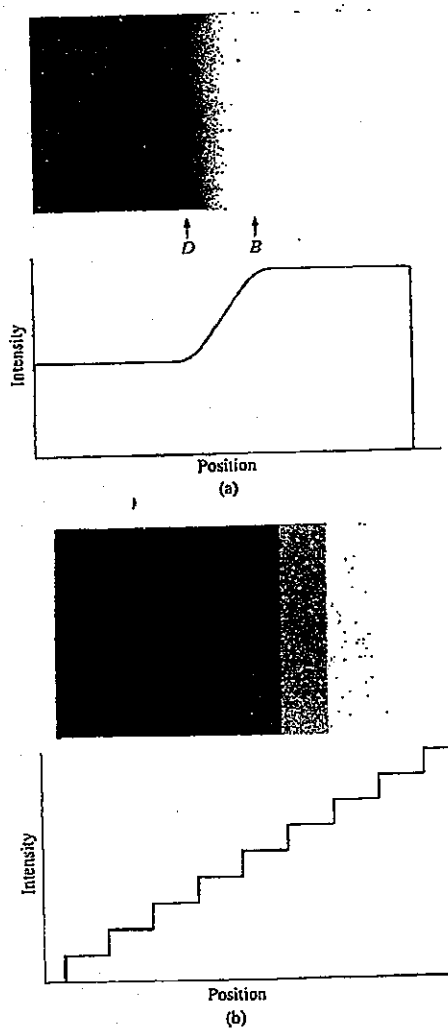


Figure 2.6 Examples of the Mach-band effect. (From Cornsweet [1970].)

where

$$0 < i(x, y) < \infty \tag{2.2-3}$$

and

$$0 < r(x, y) < 1. \tag{2.2-4}$$

Equation (2.2-4) indicates the fact that reflectance is bounded by 0 (total absorption) and 1 (total reflectance). The nature of $i(x, y)$ is determined by the light source, while $r(x, y)$ is determined by the characteristics of the objects in a scene.

The values given in Eqs. (2.2-3) and (2.2-4) are theoretical bounds. The following *average* numerical figures illustrate some typical ranges of $i(x, y)$. On a clear day, the sun may produce in excess of 9000 foot-candles of illumination on the surface of the earth. This figure decreases to less than 1000 foot-candles on a cloudy day. On a clear evening, a full moon yields about 0.01 foot-candle of illumination. The typical illumination level in a commercial office is about 100 foot-candles. Similarly, the following are some typical values of $r(x, y)$: 0.01 for black velvet, 0.65 for stainless steel, 0.80 for flat-white wall paint, 0.90 for silver-plated metal, and 0.93 for snow.

Throughout this book, the intensity of a monochrome image f at coordinates (x, y) will be called the *gray level* (l) of the image at that point. From Eqs. (2.2-2) through (2.2-4), it is evident that l lies in the range

$$L_{\min} \leq l \leq L_{\max}. \tag{2.2-5}$$

In theory, the only requirement on $L_{\min}$ is that it be positive, and on $L_{\max}$ that it be finite. In practice, $L_{\min} = i_{\min} r_{\min}$ and $L_{\max} = i_{\max} r_{\max}$. Using the above values of illumination and reflectance as a guideline, one may expect the values $L_{\min} \approx 0.005$ and $L_{\max} = 100$ for indoor image processing applications.

The interval $[L_{\min}, L_{\max}]$ is called the *gray scale*. It is common practice to shift this interval numerically to the interval $[0, L]$, where $l = 0$ is considered black and $l = L$ is considered white in the scale. All intermediate values are shades of gray varying continuously from black to white.

2.3 SAMPLING AND QUANTIZATION

2.3.1 Uniform Sampling and Quantization

In order to be in a form suitable for computer processing, an image function $f(x, y)$ must be digitized both spatially and in amplitude. Digitization of the spatial coordinates (x, y) will be referred to as *image sampling*, while amplitude digitization will be called *gray-level quantization*.

Suppose that a continuous image $f(x, y)$ is approximated by equally spaced samples arranged in the form of an $N \times N$ array† as shown in Eq. (2.3-1), where

† Digitization of an image need not be limited to square arrays. However, following discussions will often be simplified by the adoption of this convention.

each element of the array is a discrete quantity:

$$f(x, y) \approx \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1) & \cdots & f(1,N-1) \\ \vdots & & & \\ f(N-1,0) & f(N-1,1) & \cdots & f(N-1,N-1) \end{bmatrix} \quad (2.3-$$

The right side of this equation represents what is commonly called a *digital imag* while each element of the array is referred to as an *image element, picture eleme* *pixel,* or *pel,* as indicated in Section 1.2. The terms "image" and "pixels" w be used throughout the following discussions to denote a digital image and its elemen

The above digitization process requires that a decision be made on a value f N as well as on the number of discrete gray levels allowed for each pixel. It common practice in digital image processing to let these quantities be integer pow of two; that is,

$$N = 2^n \quad (2.3-$$

and

$$G = 2^m, \quad (2.3$$

where G denotes the number of gray levels. It is assumed in this section that discrete levels are equally spaced between 0 and L in the gray scale. Using E (2.3-2) and (2.3-3) the number, b, of bits required to store a digitized image given by

$$b = N \times N \times m. \quad (2.3$$

Table 2.1 Number of Storage Bits for Various Values of N and m

N \ m	1	2	3	4	5	6	7	8
32	1,024	2,048	3,072	4,096	5,120	6,144	7,168	8
64	4,096	8,192	12,288	16,384	20,480	24,576	28,672	32
128	16,384	32,768	49,152	65,536	81,920	98,304	114,688	131
256	65,536	131,072	196,608	262,144	327,680	393,216	458,752	524
512	262,144	524,288	786,432	1,048,576	1,310,720	1,572,864	1,835,008	2,097

Table 2.2 Number of 8-bit Bytes of Storage for Various Values of N and m

N \ m	1	2	3	4	5	6	7	8
32	128	256	512	512	1,024	1,024	1,024	1,024
64	512	1,024	2,048	2,048	4,096	4,096	4,096	4,096
128	2,048	4,096	8,192	8,192	16,384	16,384	16,384	16,384
256	8,192	16,384	32,768	32,768	65,536	65,536	65,536	65,536
512	32,768	65,536	131,072	131,072	262,144	262,144	262,144	262,144

For example, a 128 × 128 image with 64 gray levels requires 98,304 bits of storage. Table 2.1 summarizes values of b for some typical ranges of N and m. Table 2.2 gives the corresponding number of 8-bit bytes. Generally, it is not practical from a programming point of view to fill a byte completely if this implies a pixel overlap from one byte to the next. Thus the figures in Table 2.2 represent the minimum number of bytes needed for each value of N and m when no overlap is allowed. For example, if $m = 5$, it is assumed that only one pixel is stored in a byte, even though this leaves three unused bits in the byte.

Since Eq. (2.3-1) is an approximation to a continuous image, a reasonable question to ask at this point is how many samples and gray levels are required for a good approximation. The *resolution* (i.e., the degree of discernible detail) of an image is strongly dependent on both N and m. The more these parameters are increased, the closer the digitized array will approximate the original image. However, Eq. (2.3-4) clearly points out the unfortunate fact that storage and, consequently, processing requirements increase rapidly as a function of N and m.

In view of the above comments, it is of interest to consider the effect that variations in N and m have on image quality. As might be suspected, a "good" image is difficult to define because quality requirements vary according to application. Figure 2.7 shows the effect of reducing the sampling-grid size on an image. Figure 2.7(a) is a 512 × 512, 256-level image showing Astronaut Buzz Aldrin during the first moon landing (note the reflection of Neil Armstrong on the face plate). Figures 2.7(b) through (f) show the same image, but with $N = 256$, 128, 64, 32, and 16. In all cases the maximum number of allowed gray levels was kept at 256. Since the display area used for each image was the same (i.e., 512 × 512 display points), pixels in the lower-resolution images were duplicated in order to fill the entire display field. This produced a checkerboard effect, which is particularly noticeable in the low-resolution images. It is noted that the 256 × 256 image is reasonably close to Fig. 2.7(a), but image quality deteriorated rapidly for the other values of N.

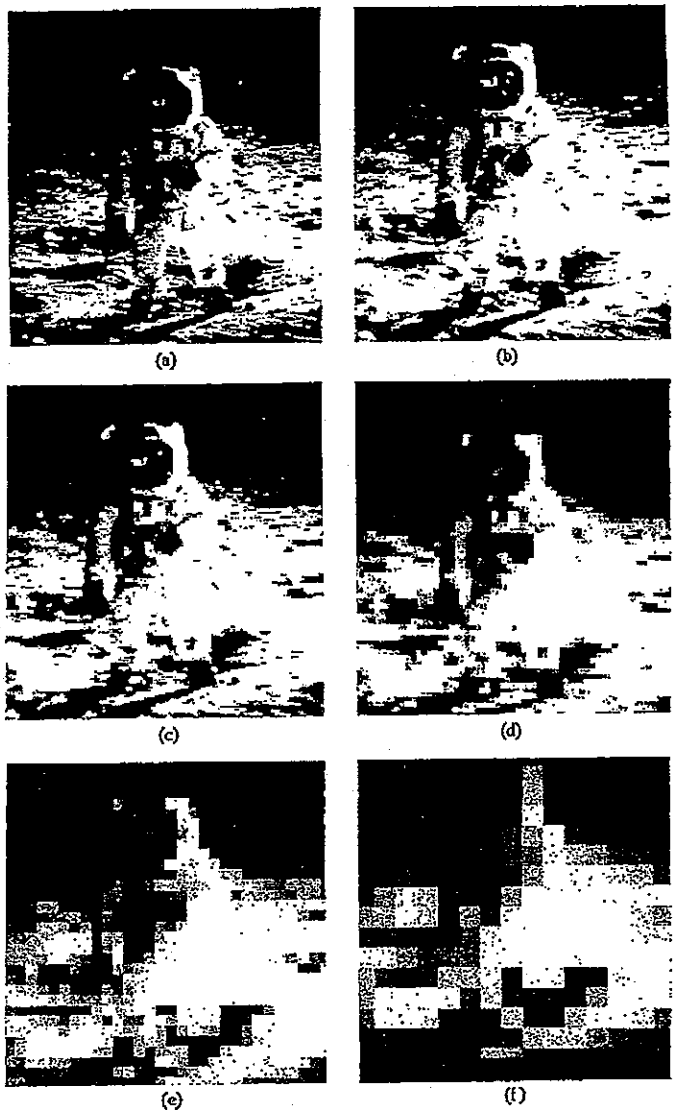


Figure 2.7 Effects of reducing sampling-grid size.

Figure 2.8 illustrates the effects produced by reducing the number of bits used to represent the gray levels in an image. Figure 2.8(a) is a picture of a subject digitized using a 512 × 512 array and 256 levels [$m = 8$ in Eq. (2.3-3)]. Figures 2.8(b) through (h) were obtained by reducing the number of bits from $m = 7$ to $m = 1$, respectively, while keeping the digitizing grid at 512 × 512. The 256-, 128-, and 64-level images are of acceptable quality. The 32-level image, however, has some mild "false contouring" in the smooth background area above the subject's right shoulder. This effect is considerably more pronounced in the image displayed in 16 levels, and increases sharply for the remaining images.

The number of samples and gray levels required to produce a faithful reproduction of an original image depends on the image itself. As a basis for comparison, the requirements to obtain a quality comparable to that of monochrome TV pictures over a wide range of image types are on the order of 512 × 512 pixels with 128 gray levels. As a rule, a minimum system for general image processing work should be able to display 256 × 256 pixels with 64 gray levels.

The above results illustrate the effects produced on image quality by varying N and m independently. However, these results only partially answer the question posed earlier since nothing has yet been said about the relation between these parameters. Huang [1965] considered this problem in an attempt to quantify experimentally the effects on image quality produced by varying N and m. The experiment consisted of a set of subjective tests. Three of the images used are shown in Fig. 2.9 on page 28. The woman's face is representative of an image with relatively little detail; the picture of the cameraman contains an intermediate amount of detail; and the crowd picture contains, by comparison, a large amount of detail information.

Sets of these three images were generated by varying N and m and observers were then asked to rank them according to their subjective quality. The results are summarized in Fig. 2.10 (page 29) in the form of *isopreference curves* in the N-m plane. Each point in this plane represents an image with values of N and m equal to the coordinates of that point. An isopreference curve is one in which the points represent images of equal subjective quality.

The isopreference curves of Fig. 2.10 are arranged, from left to right, in order of increasing subjective quality. These results suggest several empirical conclusions: (1) As expected, the quality of the images tends to increase as N and m are increased. There were a few cases in which, for fixed N, the quality improved by decreasing m. This is most likely due to the fact that a decrease in m generally increases the apparent contrast of an image. (2) The curves tend to become more vertical as the detail in the image increases. This suggests that for images with a large amount of detail only a few gray levels are needed. For example, it is noted in Fig. 2.10(c) that, for $N = 64$ or 128, image quality is not improved by an increase in m. This is not true for the curves in the other two figures. (3) The isopreference curves depart markedly from the curves of constant b, which are shown dotted in Fig. 2.10.

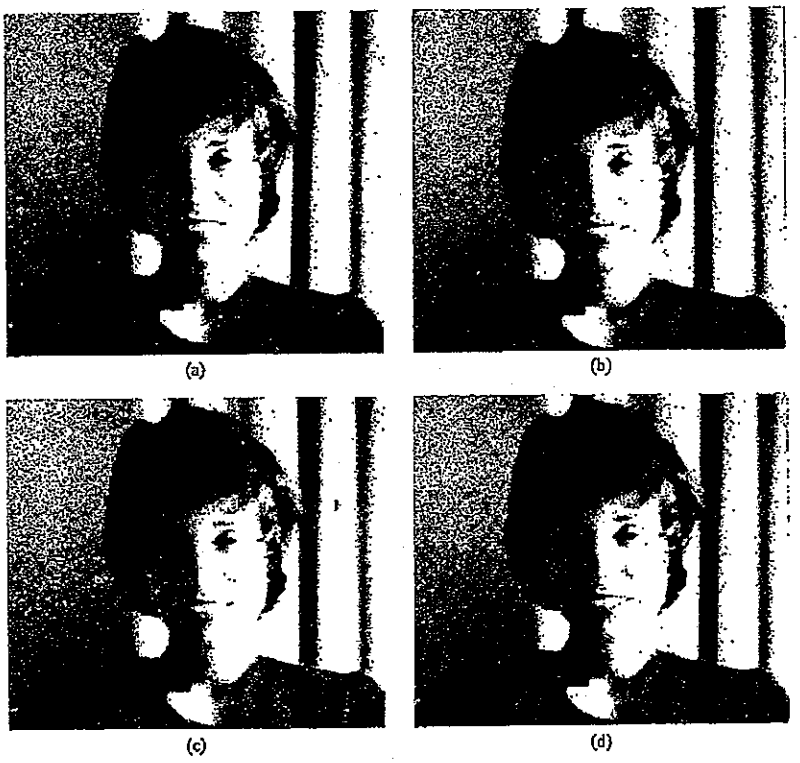


Figure 2.8 A 512 × 512 image displayed in 256, 128, 64, 32, 16, 8, 4, and 2 levels, respectively.

2.3.2 Nonuniform Sampling and Quantization

For a fixed value of N, it is possible in many cases to improve the appearance of an image by using an adaptive scheme where the sampling process depends on the characteristics of the image. In general, fine sampling is required in the neighborhood of sharp gray-level transitions, while coarse sampling may be employed in relatively smooth regions. Consider, for example, a simple image consisting of a face superimposed on a uniform background. Clearly, the background carries little detail information and can be quite adequately represented by coarse sampling. The face, on the other hand, contains considerably more detail. If the additional samples not used in the background are used in this region of the image the overall result will tend to improve, particularly if N is small. In distributing the samples, greater sample

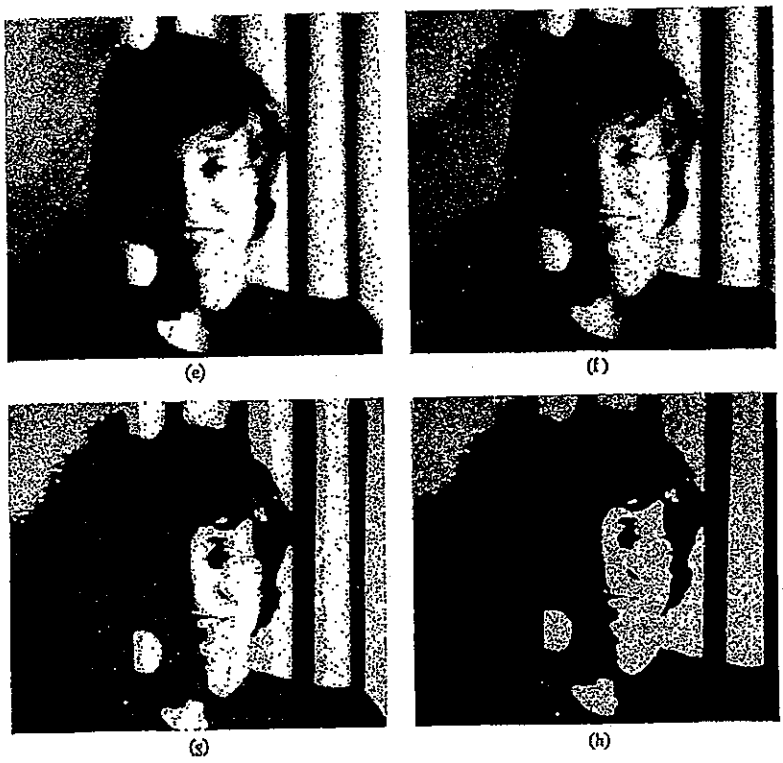


Figure 2.8 (Continued.)

concentration should be used in gray-level transition boundaries, such as the boundary between the face and the background in the preceding example.

It is important to note that the necessity of having to identify boundaries, even if only on a rough basis, is a definite drawback of the nonuniform sampling approach. Also, it should be kept in mind that this method is not practical for images containing relatively small uniform regions. For instance, nonuniform sampling would be difficult to justify for an image of a dense crowd of people.

When the number of gray levels must be kept small, it is usually desirable to use unequally spaced levels in the quantization process. A method similar to the nonuniform sampling technique discussed above may be used for the distribution of gray levels in an image. However, since the eye is relatively poor at estimating shades of gray near abrupt level changes, the approach in this case is to use few

Figure 2.9 Test images used in evaluating subjective image quality. (From Huang [1965].)

gray levels in the neighborhood of boundaries. The remaining levels can then be used in regions where gray-level variations are smooth, thus avoiding or reducing the false contours that often appear in these regions if they are too coarsely quantized.

This method is subject to the same observations made above regarding boundary detection and detail content. An alternative technique that is particularly attractive for distributing gray levels consists of computing the frequency of occurrence of all allowed levels. If gray levels in a certain range occur frequently, while others occur rarely, the quantization levels are finely spaced in this range and coarsely spaced outside of it. This method is sometimes called *tapered quantization.*

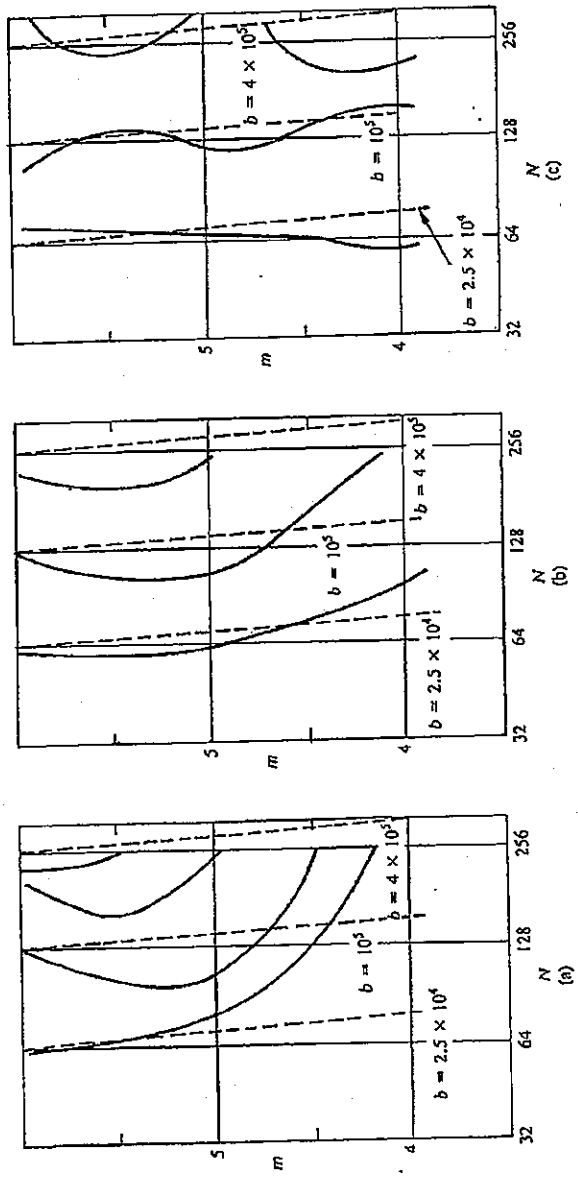


Figure 2.10 Isopreference curves for (a) face, (b) cameraman, and (c) crowd. (From Huang [1965].)

2.4 SOME BASIC RELATIONSHIPS BETWEEN PIXELS

In this section we consider several primitive, but important relationships between pixels in a digital image. As in the previous sections, an image will be denoted by $f(x, y)$. When referring to a particular pixel, we will use lowercase letters, such as p and q. A subset of pixels of $f(x, y)$ will be denoted by S.

2.4.1 Neighbors of a Pixel

A pixel p at coordinates (x, y) has four *horizontal* and *vertical* neighbors whose coordinates are given by

$$(x + 1, y), (x - 1, y), (x, y + 1), (x, y - 1).$$

This set of pixels, called the 4-*neighbors* of p, will be denoted by $N_4(p)$. It is noted that each of these pixels is a unit distance from (x, y) and also that some of the neighbors of p will be outside the digital image if (x, y) is on the border of the image.

The four *diagonal* neighbors of p have coordinates

$$(x + 1, y + 1), (x + 1, y - 1), (x - 1, y + 1), (x - 1, y - 1)$$

and will be denoted $N_D(p)$. These points, together with the 4-neighbors defined above, are called the 8-*neighbors* of p, denoted $N_8(p)$. As before, some of the points in $N_D(p)$ and $N_8(p)$ will be outside the image if (x, y) is on the border of the image.

2.4.2 Connectivity

Connectivity between pixels is an important concept used in establishing boundaries of objects and components of regions in an image. To establish whether two pixels are connected we must determine if they are adjacent in some sense (e.g., if they are 4-neighbors) and if their gray levels satisfy a specified criterion of similarity (e.g., if they are equal). For instance, in a binary image with values 0 and 1, two pixels may be 4-neighbors, but they are not said to be connected unless they have the same value.

Let V be the set of gray-level values used to define connectivity; for example, if only connectivity of pixels with intensities of 59, 60, and 61 is important, then $V = \{59, 60, 61\}$. We consider three types of connectivity:

(a) 4-*connectivity*. Two pixels p and q with values from V are 4-connected if q is in the set $N_4(p)$.

(b) 8-*connectivity*. Two pixels p and q with values from V are 8-connected if q is in the set $N_8(p)$.

(c) m-*connectivity* (mixed connectivity). Two pixels p and q with values from V are m-connected if

(i) q is in $N_4(p)$, or

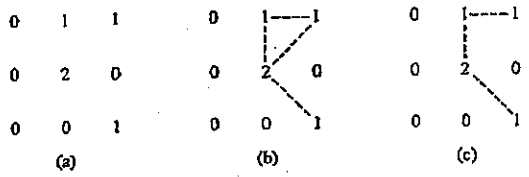


Figure 2.11 (a) Arrangement of pixels. (b) 8-neighbors of the pixel labeled "2." (c) m-neighbors of the same pixel. The dashed lines are paths between that pixel and its neighbors.

(ii) q is in $N_D(p)$ and the set $N_4(p) \cap N_4(q)$ is empty. (This is the set of pixels that are 4-neighbors of both p and q and whose values are from V.)

Mixed connectivity is a modification of 8-connectivity and is introduced to eliminate the multiple path connections (see below) that often arise when 8-connectivity is used. For example, consider the pixel arrangement shown in Fig. 2.11(a). Assuming $V = \{1, 2\}$, the paths between 8-neighbors of the pixel with value 2 are shown by dashed lines in Fig. 2.11(b). It is important to note the ambiguity in path connections that results from allowing 8-connectivity. This ambiguity is removed by using m-connectivity, as shown in Fig. 2.11(c).

A pixel p is *adjacent* to a pixel q if they are connected. We may define 4-, 8-, or m-adjacency, depending on the type of connectivity specified. Two image subsets S_1 and S_2 are adjacent if some pixel in S_1 is adjacent to some pixel in S_2.

A *path* from pixel p with coordinates (x, y) to pixel q with coordinates (s, t) is a sequence of distinct pixels with coordinates

$$(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n),$$

where $(x_0, y_0) = (x, y)$ and $(x_n, y_n) = (s, t)$, (x_i, y_i) is adjacent to (x_{i-1}, y_{i-1}), $1 \leq i \leq n$, and n is the *length* of the path. We may define 4-, 8-, or m-paths, depending on the type of adjacency used.

If p and q are pixels of an image subset S then p is *connected* to q in S if there is a path from p to q consisting entirely of pixels in S. For any pixel p in S, the set of pixels in S that are connected to p is called a *connected component* of S. It then follows that any two pixels of a connected component are connected to each other, and that distinct connected components are disjoint.

2.4.3 Distance Measures

Given pixels p, q, and z, with coordinates (x, y), (s, t), and (u, v) respectively, we call D a *distance function* or *metric* if

(a) $D(p, q) \geq 0$ ($D(p, q) = 0$ iff $p = q$),

(b) $D(p, q) = D(q, p)$,

(c) $D(p, z) \leq D(p, q) + D(q, z)$.

The *Euclidean distance* between p and q is defined as

$$D_e(p, q) = [(x - s)^2 + (y - t)^2]^{1/2}. \tag{2.4-1}$$

For this distance measure, the pixels having a distance less than or equal to some value r from (x, y) are the points contained in a disk of radius r centered at (x, y).

The D_4 *distance* (also called *city-block distance*) between p and q is defined as

$$D_4(p, q) = |x - s| + |y - t|. \tag{2.4-2}$$

In this case the pixels having a D_4 distance from (x, y) less than or equal to some value r form a diamond centered at (x, y). For example, the pixels with D_4 distance ≤ 2 from (x, y) (the center point) form the following contours of constant distance:

$$\begin{array}{ccccc} & & 2 & & \\ & 2 & 1 & 2 & \\ 2 & 1 & 0 & 1 & 2 \\ & 2 & 1 & 2 & \\ & & 2 & & \end{array}$$

It is noted that the pixels with $D_4 = 1$ are the 4-neighbors of (x, y).

The D_8 *distance* (also called *chessboard distance*) between p and q is defined as

$$D_8(p, q) = \max(|x - s|, |y - t|). \tag{2.4-3}$$

In this case the pixels with D_8 distance from (x, y) less than or equal to some value r form a square centered at (x, y). For example, the pixels with D_8 distance ≤ 2 from (x, y) (the center point) form the following contours of constant distance:

$$\begin{array}{ccccc} 2 & 2 & 2 & 2 & 2 \\ 2 & 1 & 1 & 1 & 2 \\ 2 & 1 & 0 & 1 & 2 \\ 2 & 1 & 1 & 1 & 2 \\ 2 & 2 & 2 & 2 & 2 \end{array}$$

It is noted that the pixels with $D_8 = 1$ are the 8-neighbors of (x, y).

It is also of interest to note that the D_4 distance between two points p and q is equal to the length of the shortest 4-path between these two points. Similar comments apply to the D_8 distance. In fact, we can consider both the D_4 and D_8 distances between p and q regardless of whether or not a connected path exists between them, since the definition of these distances involve only the coordinates of these points. When dealing with m-connectivity, however, the value of the distance (length

of the path) between two pixels depends on the values of the pixels along the path as well as their neighbors. For instance, consider the following arrangement of pixels, where it is assumed that p, p_2, and p_4 have a value of 1 and p_1 and p_3 have a value of 0 or 1:

$$\begin{matrix} & p_3 & p_4 \\ p_1 & p_2 & \\ p & & \end{matrix}$$

If we only allow connectivity of pixels valued 1, and p_1 and p_3 are 0, the m distance between p and p_4 is 2. If either p_1 or p_3 is 1, the distance is 3. If both p_1 and p_3 are 1, the distance is 4.

2.4.4 Arithmetic/Logic Operations

Arithmetic and logic operations between pixels are used extensively in most branches of image processing. The arithmetic operations between two pixels p and q are denoted as follows:

$$\begin{aligned} \text{Addition:} &\quad p + q \\ \text{Subtraction:} &\quad p - q \\ \text{Multiplication:} &\quad p * q \text{ (also, } pq \text{ and } p \times q) \\ \text{Division:} &\quad p \div q \end{aligned}$$

Often, one of the pixels is a constant operand, as in the multiplication of an image by a constant.

The principal logic operations used in image processing are AND, OR, and COMPLEMENT, denoted by

$$\begin{aligned} \text{AND:} &\quad p\text{AND}q \text{ (also, } p \cdot q) \\ \text{OR:} &\quad p\text{OR}q \text{ (also, } p + q) \\ \text{COMPLEMENT:} &\quad \text{NOT}q \text{ (also, } \bar{q}) \end{aligned}$$

It is well known that these operations are *functionally complete* in the sense that they can be combined to form any other logic operation. We also note that logic operations apply only to binary images, while the arithmetic operations apply to multivalued pixels.

The operations described above are used for image processing in basically two ways: on a pixel-by-pixel basis, or in neighborhood-oriented operations. For example, the addition of two images is accomplished on a pixel-by-pixel basis. Neighborhood processing is typically formulated in the context of so-called *mask* operations (the terms *template, window,* and *filter* are also often used to denote a mask). The idea behind mask operations is to let the value assigned to a pixel be a function of it and its neighbors. For instance, consider the subimage area shown in Fig. 2.12(a), and suppose that we wish to replace the value of e by the average value of the

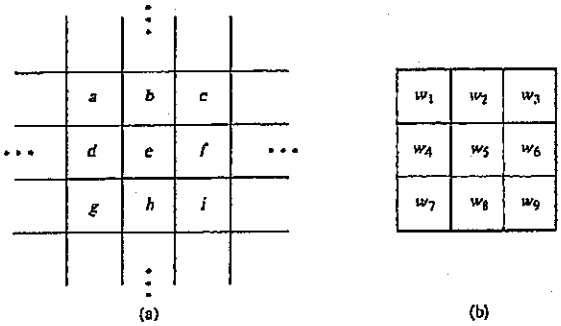


Figure 2.12 (a) Sub-area of an image showing pixel values. (b) A 3 × 3 mask with general coefficients.

pixels in a 3 × 3 region centered at e. This entails performing an arithmetic operation of the form

$$p = \frac{1}{9}\left(a + b + c + d + e + f + g + h + i\right)$$

and assigning to e the value of p.

With reference to the mask shown in Fig. 2.12(b), we may view the above operation in more general terms by centering the mask at e, multiplying each pixel under the mask by the corresponding coefficient, and adding the results; that is,

$$p = w_1a + w_2b + w_3c + w_4d + w_5e + w_6f + w_7g + w_8h + w_9i. \quad (2.4\text{-}4)$$

If we let $w_i = 1/9$, $i = 1, 2, \ldots, 9$, this operation yields the same result as the averaging procedure just discussed.

As will be seen in subsequent chapters, Eq. (2.4-4) is used widely in image processing. By properly selecting the coefficients and applying the mask at each pixel position in an image, it is possible to perform a variety of useful image operations, such as noise reduction, region thinning, and feature detection. It is noted, however, that applying a mask at each pixel location in an image is a computationally expensive task. For example, applying a 3 × 3 mask to a 512 × 512 image requires nine multiplications and eight additions at each pixel location, for a total of 2,359,296 multiplications and 2,097,152 additions.

As indicated in Section 1.3.1, most modern image processors are equipped with an Arithmetic-Logic Unit (ALU), whose function is to perform arithmetic and logic operations in parallel, typically at video-frame rates. For U.S. standard video, an ALU can perform an arithmetic or logic operation between two 512 × 512 images in 1/30th of a second. (This time interval is often called one *frame* or one *frame time*.) Given the importance of mask operations in image processing, it

is of interest to consider in some detail how to use an ALU to accelerate mask processing. For the purpose of illustration, we consider the 3 × 3 mask shown in Fig. 2.12(b) and the implementation given in Eq. (2.4-4). However, the method is easily extendible to an $n \times m$ mask and other arithmetic or logic operations.

The algorithm given here requires two image frame buffers with the capability to scroll and pan by one pixel location (see Section 1.3.1). Let frame buffer A contain the image to which the mask is to be applied. At the end of the process, B will contain the result of the operation. The reader is reminded that ALU operations are performed on all pixels in one frame time, while all buffer shifts are performed virtually instantaneously. It is assumed that all shifts are by one pixel. Letting $B = A$ initially, and using a dash to indicate no operation, we proceed as follows:

Operations on A	Operations on B
—	Multiply by w_5
Shift right	—
—	Add w_4*A
Shift down	—
—	Add w_1*A
Shift left	—
—	Add w_2*A
Shift left	—
—	Add w_3*A
Shift up	—
—	Add w_6*A
Shift up	—
—	Add w_9*A
Shift right	—
—	Add w_8*A
Shift right	—
—	Add w_7*A
Shift left	—
Shift down	—

The last two shifts are required because at the end of the last operation on B the images are in a position equivalent to having the mask with its w_7 coefficient over the e position. The two shifts correct this misalignment.

The key to understanding the foregoing procedure is to examine what happens in a single pixel of B by considering how a mask would have to be shifted in order to produce the result of Eq. (2.4-4) in that location. The first operation on B produces w_5 multiplied by the pixel value at that location. Calling that location e, we have w_5e after this operation. The first shift to the right brings neighbor d (see

Fig. 2.12(a)) over that location. The next operation multiplies d by w_4 and adds the result to the location of the first step. So at this point we have $w_4d + w_5e$ at the location in question. The next shift on A and ALU operation on B produce $w_1a + w_4d + w_5e$ at that location, and so on. Since the operations are done in parallel for all locations in B, the procedure just explained takes place simultaneously at the other locations in that image. In most ALUs, the operation of multiplying an image by a constant (i.e., w_i*A) followed by an ADD is done in one frame time. Thus the ALU implementation of Eq. (2.4-4) for an entire image takes on the order of nine frame times (9/30th of a second). For an $n \times m$ mask it would take on the order of nm frame times.

2.5 IMAGING GEOMETRY

In the following discussion we consider several important transformations used in imaging, derive a camera model, and treat the stereo imaging problem in some detail.

2.5.1 Some Basic Transformations

The material in this section deals with the development of a unified representation for problems such as image rotation, scaling, and translation. All transformations are expressed in a three-dimensional (3D) Cartesian coordinate system in which a point has coordinates denoted by (X, Y, Z). In cases involving two-dimensional images, we will adhere to our previous convention of using the lowercase representation (x, y) to denote the coordinates of a pixel. It is common terminology to refer to (X, Y, Z) as the *world coordinates* of a point.

Translation

Suppose that we wish to translate a point with coordinates (X, Y, Z) to a new location by using displacements (X_0, Y_0, Z_0). The translation is easily accomplished by using the following equations:

$$\begin{aligned} X^* &= X + X_0 \\ Y^* &= Y + Y_0 \\ Z^* &= Z + Z_0 \end{aligned} \tag{2.5-1}$$

where (X^*, Y^*, Z^*) are the coordinates of the new point. Equation (2.5-1) may be expressed in matrix form by writing

$$\begin{bmatrix} X^* \\ Y^* \\ Z^* \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & X_0 \\ 0 & 1 & 0 & Y_0 \\ 0 & 0 & 1 & Z_0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.5-2}$$

As indicated later in this section, it is often useful to concatenate several transformations to produce a composite result, such as translation, followed by scaling and then rotation. The notational representation of this process is simplified considerably by using square matrices. With this in mind, we may write Eq. (2.5-2) in the following form:

$$\begin{bmatrix} X^* \\ Y^* \\ Z^* \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & X_0 \\ 0 & 1 & 0 & Y_0 \\ 0 & 0 & 1 & Z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.5-3}$$

In terms of the values of X^*, Y^*, and Z^*, Eqs. (2.5-2) and (2.5-3) are equivalent.

Throughout this section, we will use the unified matrix representation

$$\mathbf{v}^* = \mathbf{A}\mathbf{v}, \tag{2.5-4}$$

where $\mathbf{A}$ is a 4×4 transformation matrix, $\mathbf{v}$ is the column vector containing the original coordinates,

$$\mathbf{v} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.5-5}$$

and $\mathbf{v}^*$ is a column vector whose components are the transformed coordinates

$$\mathbf{v}^* = \begin{bmatrix} X^* \\ Y^* \\ Z^* \\ 1 \end{bmatrix} \tag{2.5-6}$$

Using this notation, the matrix used for translation is given by

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & X_0 \\ 0 & 1 & 0 & Y_0 \\ 0 & 0 & 1 & Z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.5-7}$$

and the translation process is accomplished by using Eq. (2.5-4), so that $\mathbf{v}^* = \mathbf{T}\mathbf{v}$.

Scaling

Scaling by factors S_x, S_y, and S_z along the X, Y, and Z axes is given by the transformation matrix

$$\mathbf{S} = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.5-8}$$

Rotation

The transformations used for three-dimensional rotation are inherently more complex than the transformations discussed thus far. The simplest form of these transformations is for rotation of a point about the coordinate axes. To rotate a given point about an arbitrary point in space requires three transformations: the first translates the arbitrary point to the origin, the second performs the rotation, and the third translates the point back to its original position.

With reference to Fig. 2.13, rotation of a point about the Z coordinate axis by an angle θ is achieved by using the transformation

$$\mathbf{R}_\theta = \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.5-9}$$

The rotation angle θ is measured clockwise when looking at the origin from a point on the $+Z$ axis. It is noted that this transformation affects only the values of X and Y coordinates.

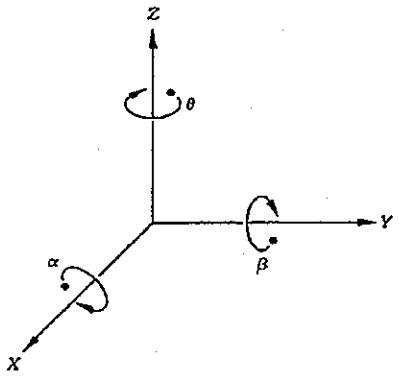


Figure 2.13 Rotation of a point about each of the coordinate axes. Angles are measured clockwise when looking along the rotation axis toward the origin.

Rotation of a point about the X axis by an angle α is performed by using the transformation

$$\mathbf{R}_\alpha = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha & 0 \\ 0 & -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.5-10}$$

Finally, rotation of a point about the Y axis by an angle β is achieved by using the transformation

$$\mathbf{R}_\beta = \begin{bmatrix} \cos\beta & 0 & -\sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.5-11}$$

Concatenation and inverse transformations

The application of several transformations can be represented by a single 4×4 transformation matrix. For example, translation, scaling, and rotation about the Z axis of a point $\mathbf{v}$ is given by

$$\begin{aligned} \mathbf{v}^* &= \mathbf{R}_\theta(\mathbf{S}(\mathbf{T}\mathbf{v})) \\ &= \mathbf{A}\mathbf{v}, \end{aligned} \tag{2.5-12}$$

where $\mathbf{A}$ is the 4×4 matrix $\mathbf{A} = \mathbf{R}_\theta\mathbf{S}\mathbf{T}$. It is important to note that these matrices generally do not commute, so the order of application is important.

Although our discussion thus far has been limited to transformations of a single point, the same ideas extend to transforming a set of m points simultaneously by using a single transformation. With reference to Eq. (2.5-5), let $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m$ represent the coordinates of m points. If we form a $4 \times m$ matrix $\mathbf{V}$ whose columns are these column vectors, then the simultaneous transformation of all these points by a 4×4 transformation matrix $\mathbf{A}$ is given by

$$\mathbf{V}^* = \mathbf{A}\mathbf{V}. \tag{2.5-13}$$

The resulting matrix $\mathbf{V}^*$ is $4 \times m$. Its ith column, $\mathbf{v}_i^*$, contains the coordinates of the transformed point corresponding to $\mathbf{v}_i$.

Before leaving this section, we point out that many of the transformations discussed above have inverse matrices that perform the opposite transformation and can be obtained by inspection. For example, the inverse translation matrix is

$$\mathbf{T}^{-1} = \begin{bmatrix} 1 & 0 & 0 & -X_0 \\ 0 & 1 & 0 & -Y_0 \\ 0 & 0 & 1 & -Z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.5-14}$$

Similarly, the inverse rotation matrix $\mathbf{R}_\theta^{-1}$ is given by

$$\mathbf{R}_\theta^{-1} = \begin{bmatrix} \cos(-\theta) & \sin(-\theta) & 0 & 0 \\ -\sin(-\theta) & \cos(-\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.5-15}$$

The inverses of more-complex transformation matrices are usually obtained by numerical techniques.

2.5.2 Perspective Transformations

A perspective transformation (also called an imaging transformation) projects 3D points onto a plane. Perspective transformations play a central role in image processing because they provide an approximation to the manner in which an image is formed by viewing a three-dimensional world. These transformations are fundamentally different from those discussed in the previous section because they are nonlinear in that they involve division by coordinate values.

A model of the image formation process is shown in Fig. 2.14. We define the camera coordinate system (x, y, z) as having the image plane coincident with the xy plane and the optical axis (established by the center of the lens) along the z axis. Thus the center of the image plane is at the origin, and the center of the lens is at coordinates $(0, 0, \lambda)$. If the camera is in focus for distant objects, λ is the *focal length* of the lens. In this section, it is assumed that the camera coordinate system is aligned with the world coordinate system (X, Y, Z). This restriction will be removed in the following section.

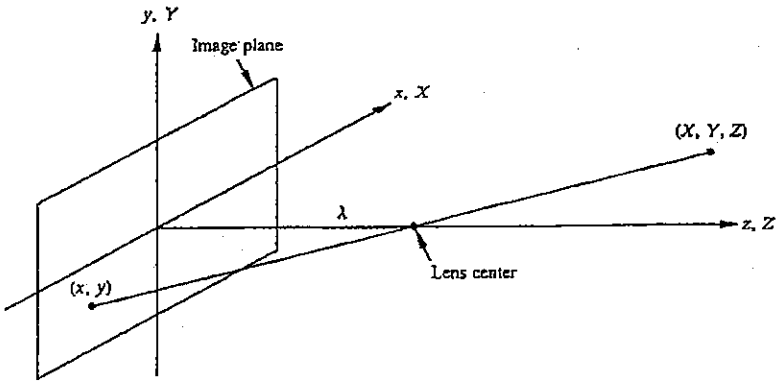


Figure 2.14 Basic model of the imaging process. The camera coordinate system (x, y, z) is aligned with the world coordinate system (X, Y, Z).

Let (X, Y, Z) be the world coordinates of any point in a 3D scene, as shown in Fig. 2.14. It will be assumed throughout the following discussion that $Z > \lambda$, that is, all points of interest lie in front of the lens. What we wish to do first is obtain a relationship that gives the coordinates (x, y) of the projection of the point (X, Y, Z) onto the image plane. This is easily accomplished by the use of similar triangles. With reference to Fig. 2.14, it follows that

$$\frac{x}{\lambda} = -\frac{X}{Z-\lambda} = \frac{X}{\lambda - Z} \tag{2.5-16}$$

and

$$\frac{y}{\lambda} = -\frac{Y}{Z-\lambda} = \frac{Y}{\lambda - Z}, \tag{2.5-17}$$

where the negative signs in front of X and Y indicate that image points are actually inverted, as can be seen from the geometry of Fig. 2.14.

The image-plane coordinates of the projected 3D point follow directly from Eqs. (2.5-16) and (2.5-17):

$$x = \frac{\lambda X}{\lambda - Z} \tag{2.5-18}$$

and

$$y = \frac{\lambda Y}{\lambda - Z}. \tag{2.5-19}$$

It is important to note that these equations are nonlinear because they involve division by the variable Z. Although we could use them directly as shown above, it is often convenient to express these equations in linear matrix form as we did in the previous section for rotation, translation, and scaling. This can be accomplished easily by using homogeneous coordinates.

The homogeneous coordinates of a point with Cartesian coordinates (X, Y, Z) are defined as (kX, kY, kZ, k), where k is an arbitrary, nonzero constant. Clearly, conversion of homogeneous coordinates back to Cartesian coordinates is accomplished by dividing the first three homogeneous coordinates by the fourth. A point in the Cartesian world coordinate system may be expressed in vector form as

$$\mathbf{w} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{2.5-20}$$

and its homogeneous counterpart is given by

$$\mathbf{w}_h = \begin{bmatrix} kX \\ kY \\ kZ \\ k \end{bmatrix} \tag{2.5-21}$$

If we define the *perspective transformation matrix*

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{-1}{\lambda} & 1 \end{bmatrix} \tag{2.5-22}$$

then the product $\mathbf{P}\mathbf{w}_h$ yields a vector that we shall denote by $\mathbf{c}_h$:

$$\begin{aligned} \mathbf{c}_h &= \mathbf{P}\mathbf{w}_h \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{-1}{\lambda} & 1 \end{bmatrix} \begin{bmatrix} kX \\ kY \\ kZ \\ k \end{bmatrix} \\ &= \begin{bmatrix} kX \\ kY \\ kZ \\ \frac{-kZ + k}{\lambda} \end{bmatrix} \end{aligned} \tag{2.5-23}$$

The elements of $\mathbf{c}_h$ are the camera coordinates in homogeneous form. As indicated above, these coordinates can be converted to Cartesian form by dividing each of the first three components of $\mathbf{c}_h$ by the fourth. Thus the Cartesian coordinates of any point in the camera coordinate system are given in vector form by

$$\mathbf{c} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \frac{\lambda X}{\lambda - Z} \\ \frac{\lambda Y}{\lambda - Z} \\ \frac{\lambda Z}{\lambda - Z} \end{bmatrix} \tag{2.5-24}$$

The first two components of $\mathbf{c}$ are the (x, y) coordinates in the image plane of a projected 3D point (X, Y, Z), as shown earlier in Eqs. (2.5-18) and (2.5-19). The third component is of no interest to us in terms of the model in Fig. 2.14. As will be seen below, this component acts as a free variable in the inverse perspective transformation.

The inverse perspective transformation maps an image point back into 3D. Thus from Eq. (2.5-23),

$$\mathbf{w}_h = \mathbf{P}^{-1}\mathbf{c}_h, \tag{2.5-25}$$

where $\mathbf{P}^{-1}$ is easily found to be

$$\mathbf{P}^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{\lambda} & 1 \end{bmatrix} \tag{2.5-26}$$

Suppose that a given image point has coordinates $(x_0, y_0, 0)$, where the 0 in the z location simply indicates the fact that the image plane is located at $z = 0$. This point may be expressed in homogeneous vector form as

$$\mathbf{c}_h = \begin{bmatrix} kx_0 \\ ky_0 \\ 0 \\ k \end{bmatrix} \tag{2.5-27}$$

Application of Eq. (2.5-25) then yields the homogeneous world coordinate vector

$$\mathbf{w}_h = \begin{bmatrix} kx_0 \\ ky_0 \\ 0 \\ k \end{bmatrix} \tag{2.5-28}$$

or, in Cartesian coordinates,

$$\mathbf{w} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \\ 0 \end{bmatrix} \tag{2.5-29}$$

This is obviously not what one would expect since it gives $Z = 0$ for *any* 3D point. The problem here is caused by the fact that mapping a 3D scene onto the image plane is a many-to-one transformation. The image point (x_0, y_0) corresponds to the set of collinear 3D points that lie on the line that passes through $(x_0, y_0, 0)$ and $(0, 0, \lambda)$. The equations of this line in the world coordinate system are obtained

from Eqs. (2.5-18) and (2.5-19); that is,

$$X = \frac{x_0}{\lambda}(\lambda - Z) \tag{2.5-30}$$

and

$$Y = \frac{y_0}{\lambda}(\lambda - Z). \tag{2.5-31}$$

These equations show that unless we know something about the 3D point that generated a given image point (for example, its Z coordinate), we cannot completely recover the 3D point from its image. This observation, which certainly is not unexpected, can be used to formulate the inverse perspective transformation by using the z component of $\mathbf{c}_h$ as a free variable, instead of 0. Thus letting

$$\mathbf{c}_h = \begin{bmatrix} kx_0 \\ ky_0 \\ kz \\ k \end{bmatrix} \tag{2.5-32}$$

we now have from Eq. (2.5-25) that

$$\mathbf{w}_h = \begin{bmatrix} kx_0 \\ ky_0 \\ kz \\ \frac{kz}{\lambda} + k \end{bmatrix} \tag{2.5-33}$$

which, upon conversion to Cartesian coordinates yields

$$\mathbf{w} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \frac{\lambda x_0}{\lambda + z} \\ \frac{\lambda y_0}{\lambda + z} \\ \frac{\lambda z}{\lambda + z} \end{bmatrix} \tag{2.5-34}$$

In other words, treating z as a free variable yields the equations

$$\begin{aligned} X &= \frac{\lambda x_0}{\lambda + z} \\ Y &= \frac{\lambda y_0}{\lambda + z} \\ Z &= \frac{\lambda z}{\lambda + z} \end{aligned} \tag{2.5-35}$$

Solving for z in terms of Z in the last equation and substituting in the first two expressions yields

$$X = \frac{x_0}{\lambda}(\lambda - Z) \tag{2.5-36}$$

$$Y = \frac{y_0}{\lambda}(\lambda - Z), \tag{2.5-37}$$

which agrees with the above observation that recovering a 3D point from its image by means of the inverse perspective transformation requires knowledge of at least one of the world coordinates of the point. This problem will be addressed again in Section 2.5.5.

2.5.3 Camera Model

Equations (2.5-23) and (2.5-24) characterize the formation of an image via the projection of 3D points onto an image plane. These two equations thus constitute a basic mathematical model of an imaging camera. This model is based on the assumption that the camera and world coordinate systems are coincident. In this section we consider a more general problem in which the two coordinate systems are allowed to be separate. However, the basic objective of obtaining the image-plane coordinates of any given world point remains the same.

The situation is depicted in Fig. 2.15, which shows a world coordinate system (X, Y, Z) used to locate both the camera and 3D points (denoted by w). This figure also shows the camera coordinate system (x, y, z) and image points (denoted by c). It is assumed that the camera is mounted on a gimbal, which allows pan through an angle θ and tilt through an angle α. In this discussion, pan is defined as the angle between the x and X axes, and tilt as the angle between the z and Z axes. The offset of the center of the gimbal from the origin of the world coordinate system is denoted by vector $\mathbf{w}_0$, and the offset of the center of the imaging plane with respect to the gimbal center is denoted by a vector $\mathbf{r}$, with components (r_1, r_2, r_3).

The concepts developed in the last two sections provide all the necessary tools to derive a camera model based on the geometrical arrangement of Fig. 2.15. The approach is to bring the camera and world coordinate systems into alignment by applying a set of transformations. After this has been accomplished, we simply apply the perspective transformation given in Eq. (2.5-22) to obtain the image-plane coordinates of any given world point. In other words, we first reduce the problem to the geometrical arrangement shown in Fig. 2.14 before applying the perspective transformation.

Suppose that, initially, the camera was in *normal position*, in the sense that the gimbal center and origin of the image plane were at the origin of the world coordinate system, and all axes were aligned. Starting from normal position, the geometrical arrangement of Fig. 2.15 can be achieved in a number of ways. We

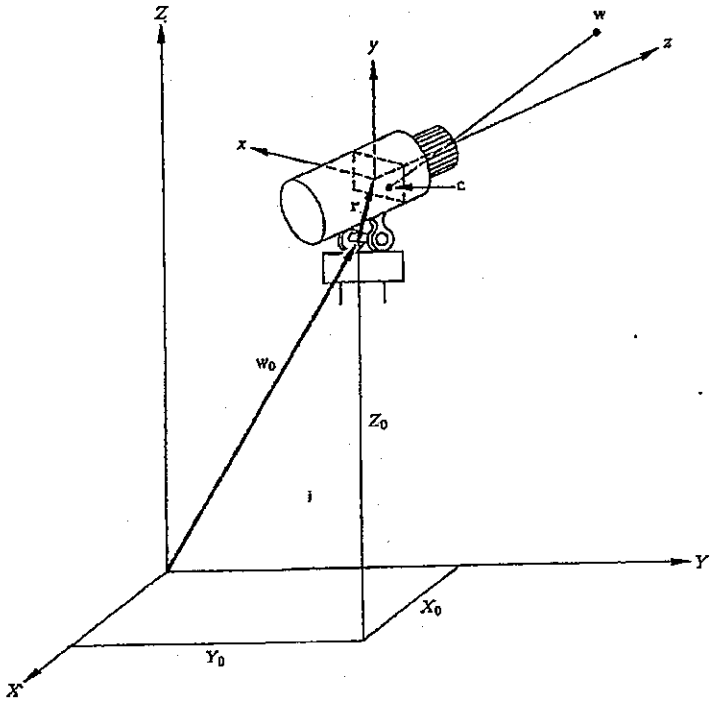


Figure 2.15 Imaging geometry with two coordinate systems. (From Fu, Gonzalez, and Lee [1987].)

assume the following sequence of steps: (1) displacement of the gimbal center from the origin, (2) pan of the x axis, (3) tilt of the z axis, and (4) displacement of the image plane with respect to the gimbal center.

Obviously, the sequence of mechanical steps just discussed does not affect the world points, since the set of points seen by the camera after it was moved from normal position is quite different. However, we can achieve normal position again simply by applying exactly the same sequence of steps to all world points. Since a camera in normal position satisfies the arrangement of Fig. 2.14 for application of the perspective transformation, our problem is thus reduced to applying to every world point a set of transformations that correspond to the steps given above.

Translation of the origin of the world coordinate system to the location of the gimbal center is accomplished by using the transformation matrix given in the follow-

ing equation:

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & -X_0 \\ 0 & 1 & 0 & -Y_0 \\ 0 & 0 & 1 & -Z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.5-38}$$

In other words, a homogeneous world point $\mathbf{w}_h$ that was at coordinates (X_0, Y_0, Z_0) is at the origin of the new coordinate system after the transformation $\mathbf{Gw}_h$.

As indicated earlier, the pan angle is measured between the x and X axes. In normal position, these two axes are aligned. In order to pan the x axis through the desired angle, we simply rotate it by θ. The rotation is with respect to the z axis and is accomplished by using the transformation matrix $\mathbf{R}_\theta$ given in Eq. (2.5-9). In other words, application of this matrix to all points (including the point $\mathbf{Gw}_h$) effectively rotates the x axis to the desired location. When using Eq. (2.5-9), it is important to keep clearly in mind the convention established in Fig. 2.13. That is, angles are considered positive when points are rotated clockwise, which implies a counterclockwise rotation of the camera about the z axis. The unrotated (0°) position corresponds to the case when the x and X axes are aligned.

At this point in the development the z and Z axes are still aligned. Since tilt is the angle between these two axes, we tilt the camera an angle α by rotating the z axis by α. The rotation is with respect to the x axis and is accomplished by applying the transformation matrix $\mathbf{R}_\alpha$ given in Eq. (2.5-10) to all points (including the point $\mathbf{R}_\theta\mathbf{Gw}_h$). As above, a counterclockwise rotation of the camera implies positive angles, and the 0° mark is when the z and Z axes are aligned.†

According to the discussion in Section 2.5.4, the two rotation matrices can be concatenated into a single matrix, $\mathbf{R} = \mathbf{R}_\alpha \mathbf{R}_\theta$. It then follows from Eqs. (2.5-9) and (2.5-10) that

$$\mathbf{R} = \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta\cos\alpha & \cos\theta\cos\alpha & \sin\alpha & 0 \\ \sin\theta\sin\alpha & -\cos\theta\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.5-39}$$

Finally, displacement of the origin of the image plane by vector $\mathbf{r}$ is achieved by the transformation matrix

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & -r_1 \\ 0 & 1 & 0 & -r_2 \\ 0 & 0 & 1 & -r_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.5-40}$$

† A useful way to visualize these transformations is to construct an axis system (e.g., with pipe cleaners), label the axes x, y, and z, and perform the rotations manually, one axis at a time.

Thus by applying to $\mathbf{w}_h$ the series of transformations $\mathbf{CRGw}_h$, we have brought the world and camera coordinate systems into coincidence. The image-plane coordinates of a point $\mathbf{w}_h$ are finally obtained by using Eq. (2.5-23). In other words, a homogeneous world point that is being viewed by a camera satisfying the geometrical arrangement shown in Fig. 2.15 has the following homogeneous representation in the camera coordinate system:

$$\mathbf{c}_h = \mathbf{PCRGw}_h. \tag{2.5-41}$$

This equation represents a perspective transformation involving two coordinate systems.

As indicated in the previous section, we obtain the Cartesian coordinates (x, y) of the imaged point by dividing the first and second components of $\mathbf{c}_h$ by the fourth. Expanding Eq. (2.5-41) and converting to Cartesian coordinates yields

$$x = \lambda \frac{(X - X_0)\cos\theta + (Y - Y_0)\sin\theta - r_1}{-(X - X_0)\sin\theta\ \sin\alpha\ + (Y - Y_0)\cos\theta\ \sin\alpha\ - (Z - Z_0)\cos\alpha + r_3 + \lambda} \tag{2.5-42}$$

and

$$y = \lambda \frac{-(X - X_0)\sin\theta\cos\alpha + (Y - Y_0)\cos\theta\cos\alpha + (Z - Z_0)\sin\alpha - r_2}{-(X - X_0)\sin\theta\ \sin\alpha\ + (Y - Y_0)\cos\theta\ \sin\alpha\ - (Z - Z_0)\cos\alpha + r_3 + \lambda} \tag{2.5-43}$$

which are the image coordinates of a point w whose world coordinates are (X, Y, Z). It is noted that these equations reduce to Eqs. (2.5-18) and (2.5-19) when $X_0 = Y_0 = Z_0 = 0$, $r_1 = r_2 = r_3 = 0$, and $\alpha = \theta = 0°$.

Example: As an illustration of the concepts just discussed, suppose that we wish to find the image coordinates of the corner of the block shown in Fig. 2.16. The camera is offset from the origin and is viewing the scene with a pan of 135° and a tilt of 135°. We will follow the convention established above that transformation angles are positive when the camera rotates in a counterclockwise manner as seen when viewing the origin along the axis of rotation.

Let us examine in detail the steps required to move the camera from normal position to the geometry shown in Fig. 2.16. The camera is shown in normal position in Fig. 2.17(a) and displaced from the origin in Fig. 2.17(b). It is important to note that, after this step, the world coordinate axes are used only to establish angle references. That is, after displacement of the world coordinate origin, all rotations take place about the new (camera) axes. Figure 2.17(c) shows a view along the z axis of the camera to establish pan. In this case the rotation of the camera about the z axis is counterclockwise, so world points are rotated about this axis in the opposite direction, which makes θ a positive angle. Figure 2.17(d) shows a view, after pan, along the x axis of the camera to establish tilt. The rotation about this axis is counterclockwise, which makes α a positive angle. The world coordinate

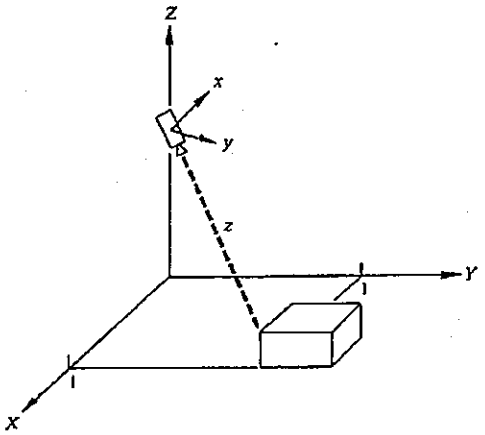


Figure 2.16 Camera viewing a 3D scene. (From Fu, Gonzalez, and Lee [1987].)

axes are shown dashed in the latter two figures to emphasize the fact that their only use is to establish the zero reference for the pan and tilt angles. We do not show in this figure the final step of displacing the image plane from the center of the gimbal.

The following parameter values apply to this problem:

$$X_0 = 0 \text{ m}$$
$$Y_0 = 0 \text{ m}$$
$$Z_0 = 1 \text{ m}$$
$$\alpha = 135°$$
$$\theta = 135°$$
$$r_1 = 0.03 \text{ m}$$
$$r_2 = r_3 = 0.02 \text{ m}$$
$$\lambda = 35 \text{ mm} = 0.035 \text{ m}$$

The corner in question is at coordinates $(X, Y, Z) = (1, 1, 0.2)$.

To compute the image coordinates of the block corner we simply substitute the above parameter values into Eqs. (2.5-42) and (2.5-43); that is,

$$x = \lambda \frac{-0.03}{-1.53 + \lambda}.$$

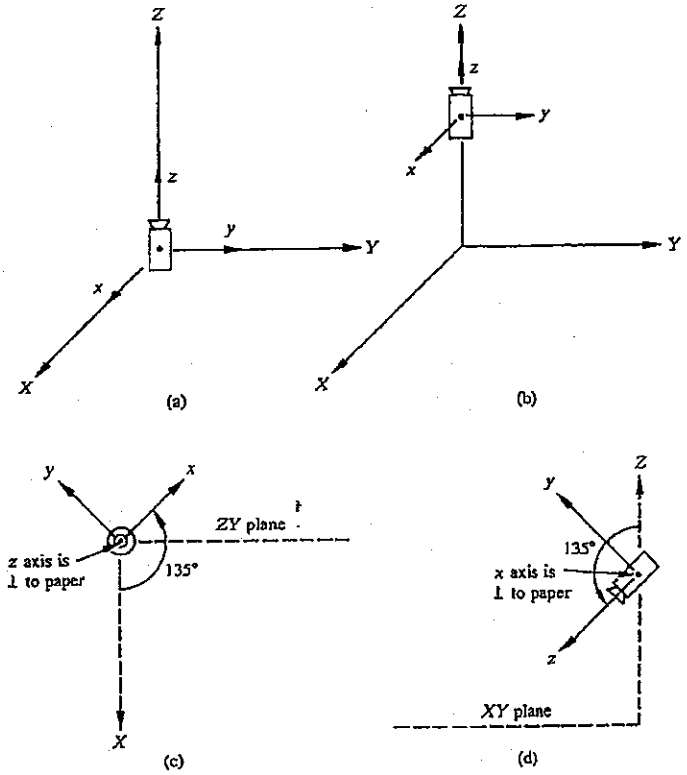


Figure 2.17 (a) Camera in normal position. (b) Gimbal center displaced from origin. (c) Observer view of rotation about z axis to determine pan angle. (d) Observer view of rotation about x axis for tilt. (From Fu, Gonzalez, and Lee [1987].)

Similarly,

$$y = \lambda \frac{-0.42}{-1.53 + \lambda}.$$

Substituting $\lambda = 0.035$ yields the image coordinates

$$x = 0.0007 \text{ m}$$

and

$$y = 0.009 \text{ m}.$$

We note that these coordinates are well within a 1 × 1 in. (0.025 × 0.025 m) imaging plane. If, for example, we had used a lens with a 200-mm focal length, it is easily verified from the above results that the corner of the block would have been imaged outside the boundary of a plane with these dimensions (i.e., it would have been outside the effective field of view of the camera).

Finally, we point out that all coordinates obtained via the use of Eqs. (2.5-42) and (2.5-43) are with respect to the center of the image plane. A change of coordinates would be required to use the convention established earlier that the origin of an image is at its top left corner. □

2.5.4 Camera Calibration

In the previous section we obtained explicit equations for the image coordinates, (x, y), of a world point w. As shown in Eqs. (2.5-42) and (2.5-43), implementation of these equations requires knowledge of the focal length, offsets, and angles of pan and tilt. While these parameters could be measured directly, it is often more convenient (e.g., when the camera moves frequently) to determine one or more of the parameters using the camera itself as a measuring device. This requires a set of image points whose world coordinates are known, and the computational procedure used to obtain the camera parameters using these known points is often referred to as *camera calibration*.

With reference to Eq. (2.5-41), let $\mathbf{A} = \mathbf{PCRG}$. The elements of $\mathbf{A}$ contain all the camera parameters, and we know from Eq. (2.5-41) that $\mathbf{c}_h = \mathbf{A}\mathbf{w}_h$. Letting $k = 1$ in the homogeneous representation, we may write

$$\begin{bmatrix} c_{h1} \\ c_{h2} \\ c_{h3} \\ c_{h4} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.5-44}$$

From the discussion in the previous two sections we know that the camera coordinates in Cartesian form are given by

$$x = c_{h1}/c_{h4} \tag{2.5-45}$$

and

$$y = c_{h2}/c_{h4}. \tag{2.5-46}$$

Substituting $c_{h1} = xc_{h4}$ and $c_{h2} = yc_{h4}$ in Eq. (2.5-44) and expanding the matrix product yields

$$\begin{aligned} xc_{h4} &= a_{11}X + a_{12}Y + a_{13}Z + a_{14} \\ yc_{h4} &= a_{21}X + a_{22}Y + a_{23}Z + a_{24} \\ c_{h4} &= a_{41}X + a_{42}Y + a_{43}Z + a_{44}, \end{aligned} \tag{2.5-47}$$

where expansion of c_{h3} has been ignored because it is related to z.

Substitution of c_{44} in the first two equations of (2.5-47) yields two equations with 12 unknown coefficients:

$$a_{11}X + a_{12}Y + a_{13}Z - a_{41}xX - a_{42}xY - a_{43}xZ - a_{44}x + a_{14} = 0 \qquad (2.5\text{-}48)$$

$$a_{21}X + a_{22}Y + a_{23}Z - a_{41}yX - a_{42}yY - a_{43}yZ - a_{44}y + a_{24} = 0 \qquad (2.5\text{-}49)$$

The calibration procedure then consists of (1) obtaining $m \geq 6$ world points (there are *two* equations) with known coordinates (X_i, Y_i, Z_i), $i = 1, 2, \ldots, m$, (2) imaging these points with the camera in a given position to obtain the corresponding image points (x_i, y_i), $i = 1, 2, \ldots, m$, and (3) using these results in Eqs. (2.5-48) and (2.5-49) to solve for the unknown coefficients. There are many numerical techniques for finding an optimal solution to a linear system of equations such as the one given by these equations (see, for example, Noble [1969]).

2.5.5 Stereo Imaging

It was noted in Section 2.5.2 that mapping a 3D scene onto an image plane is a many-to-one transformation. That is, an image point does not uniquely determine the location of a corresponding world point. It is shown in this section that the missing *depth* information can be obtained by using stereoscopic (*stereo* for short) imaging techniques.

As shown in Fig. 2.18, stereo imaging involves obtaining two separate image views of an object of interest (a single world point w in our present discussion). The distance between the centers of the two lenses is called the *baseline*, and the

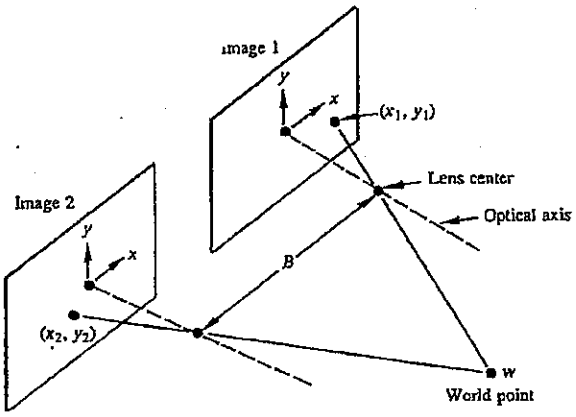


Figure 2.18 Model of the stereo imaging process. (From Fu, Gonzalez, and Lee [1987].)

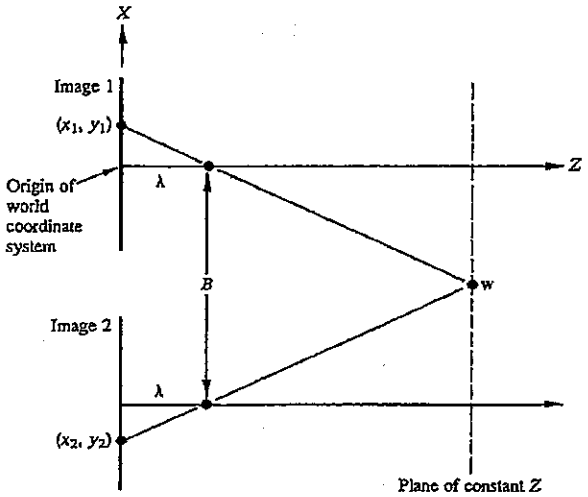


Figure 2.19 Top view of Fig. 2.18 with the first camera brought into coincidence with the world coordinate system. (From Fu, Gonzalez, and Lee [1987].)

objective is to find the coordinates (X, Y, Z) of the point w given its image points (x_1, y_1) and (x_2, y_2). It is assumed that the cameras are identical and that the coordinate systems of both cameras are perfectly aligned, differing only in the location of their origins, a condition usually met in practice. Recall our convention that, after the camera and world coordinate systems have been brought into coincidence, the xy plane of the image is aligned with the XY plane of the world coordinate system. Then, under the above assumption, the Z coordinate of w is exactly the same for both camera coordinate systems.

Suppose that we bring the first camera into coincidence with the world coordinate system, as shown in Fig. 2.19. Then, from Eq. (2.5-30), w lies on the line with (partial) coordinates

$$X_1 = \frac{x_1}{\lambda}(\lambda - Z_1), \tag{2.5-50}$$

where the subscripts on X and Z indicate that the first camera was moved to the origin of the world coordinate system, with the second camera and w following, but keeping the relative arrangement shown in Fig. 2.18. If, instead, the second camera had been brought to the origin of the world coordinate system, then we

would have that w lies on the line with (partial) coordinates

$$X_2 = \frac{x_2}{\lambda}(\lambda - Z_2). \tag{2.5-51}$$

However, due to the separation between cameras and the fact that the Z coordinate of w is the same for both camera coordinate systems, it follows that

$$X_2 = X_1 + B \tag{2.5-52}$$

and

$$Z_2 = Z_1 = Z, \tag{2.5-53}$$

where, as indicated above, B is the baseline distance.

Substitution of Eqs. (2.5-52) and (2.5-53) into Eq. (2.5-50) and (2.5-51) results in the following equations:

$$X_1 = \frac{x_1}{\lambda}(\lambda - Z) \tag{2.5-54}$$

and

$$X_1 + B = \frac{x_2}{\lambda}(\lambda - Z). \tag{2.5-55}$$

Subtracting Eq. (2.5-54) from (2.5-55) and solving for Z yields the expression

$$Z = \lambda - \frac{\lambda B}{x_2 - x_1}, \tag{2.5-56}$$

which indicates that if the difference between the corresponding image coordinates x_2 and x_1 can be determined, and the baseline and focal length are known, calculating the Z coordinate of w is a simple matter. The X and Y world coordinates then follow directly from Eqs. (2.5-30) and (2.5-31) using either (x_1, y_1) or (x_2, y_2).

The most difficult task in using Eq. (2.5-56) to obtain Z is to actually find two corresponding points in different images of the same scene. Since these points are generally in the same vicinity, a frequently used approach is to select a point within a small region in one of the image views and then attempt to find the best matching region in the other view by using correlation techniques, as discussed in Chapter 8. When the scene contains distinct features, such as prominent corners, a feature-matching approach will generally yield a faster solution for establishing correspondence.

Before leaving this discussion, we point out that the calibration procedure developed in the previous section is directly applicable to stereo imaging by simply treating the cameras independently.

2.6 PHOTOGRAPHIC FILM

Photographic film is an important element of image processing systems. It is often used as the medium where input images are recorded, and it is by far the most popular medium for recording output results. For these reasons, we conclude this chapter with a discussion of some basic properties of monochrome photographic film and their relation to image processing applications.

2.6.1 Film Structure and Exposure

A cross section of a typical photographic film as it would appear under magnification is shown in Fig. 2.20. It consists of the following layers and components: (1) a supercoat of gelatin used for protection against scratches and abrasion marks; (2) an emulsion layer consisting of minute silver halide crystals; (3) a substrate layer, which promotes adhesion of the emulsion to the film base; (4) the film base or support, made of cellulose triacetate or a related polymer; and (5) a backing layer to prevent curling.

When the film is exposed to light, the silver halide grains absorb optical energy and undergo a complex physical change. The grains that have absorbed a sufficient amount of energy contain tiny patches of metallic silver, called *development centers*. When the exposed film is developed, the existence of a single development center in a silver halide grain can precipitate the change of the entire grain to metallic silver. The grains that do not contain development centers do not undergo such a change. After development, the film is "fixed" by chemical removal of the remaining silver halide grains. The more light that reaches a given area of the film, the more silver halide is rendered developable and the denser the silver deposit that is formed there. Since the silver grains are largely opaque at optical frequencies, an image of gray tones is obtained where the brightness levels are reversed, thus producing the familiar film negative.

The process is repeated to obtain a positive picture. The negative is projected onto a sensitive paper carrying a silver halide emulsion similar to that used for the

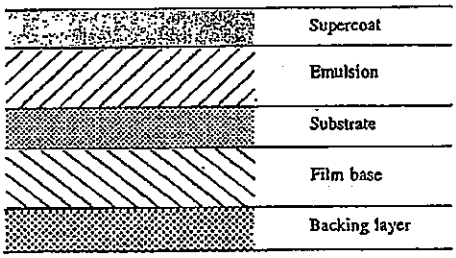


Figure 2.20 Structure of modern black-and-white film.

film. Exposure by a light source yields a latent image of the negative. After development, the paper bears a positive silver image. Enlargement of the negative is controlled by the choice of light source and size of positive paper used.

2.6.2 Film Characteristics

Of practical interest to the photographer are contrast, speed, graininess, and resolving power. An understanding of the effect of these parameters is particularly important in specialized applications such as photographing the results obtained in an image processing system.

Contrast

High-contrast films reproduce tone differences in the subject as large density differences in the photograph; low-contrast films translate tone differences as small density differences. The exposure E to which a film is subjected is defined as *energy per unit area* at each point on the photosensitive area. Exposure depends on the incident intensity I and the duration of the exposure T. These quantities are related by the expression

$$E = IT. \tag{2.6-1}$$

The most widely used description of the photosensitive properties of photographic film is a plot of the density of the silver deposit on a film versus the logarithm of E. These curves are called characteristic curves, D-log-E curves (density vs. log exposure), and H & D curves (after Hurter and Driffield, who developed the method). Figure 2.21 shows a typical H & D curve for a photographic negative. When the

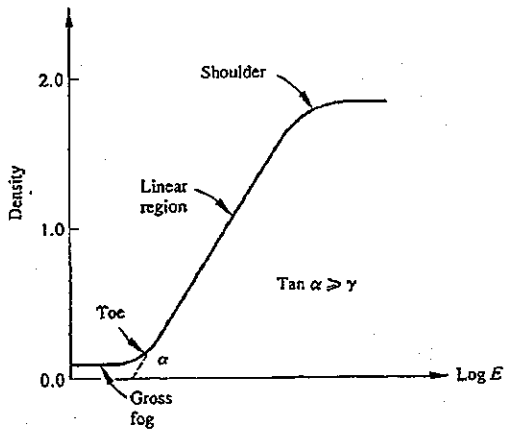


Figure 2.21 A typical H & D curve.

exposure is below a certain level, the density is independent of exposure and equal to a minimum value called the *gross fog*. In the *toe* of the curve, density begins to increase with increasing exposure. There follows a region of the curve in which density is linearly proportional to logarithmic exposure. The slope of this linear region is referred to as the film *gamma* (γ). Finally, the curve saturates in a region called the *shoulder*, and again there is no change in density with increasing exposure. The value of γ is a measure of film contrast: the steeper the slope, the higher the contrast rendered. General-purpose films of medium contrast have gammas in the range 0.7 to 1.0. High-contrast films have gammas on the order of 1.5 to 10. As a rule, films with relatively low gammas are used for continuous-tone reproduction; high-contrast films are used for copying line originals and other specialized purposes.

Speed

The speed of a film determines how much light is needed to produce a given amount of silver on development. The lower the speed, the longer the film must be exposed to record a given image. The most widely used standard of speed is the ASA scale. This scale is arithmetic, with the speed number directly proportional to the sensitivity of the film. A film of ASA 200 is twice as fast (and for a given subject requires half as much exposure) as a film of ASA 100. Some speed scales, such as the DIN system used in Europe, are logarithmic. Every increase of three in the DIN speed number doubles the actual speed. An ASA 50 film is equivalent to a DIN 18, an ASA 100 to a DIN 21, and so on.

General-purpose films for outdoor and some indoor photography have speeds between ASA 80 and ASA 160; fine-grain films for maximum image definition between ASA 20 and ASA 64; high-speed films for poor light and indoor photography between ASA 200 and ASA 500; and ultraspeed films for very low light levels from ASA 650 and up.

Graininess

The image derived from the silver halide crystals is discontinuous in structure. This gives an appearance of graininess in big enlargements. The effect is most prominent in fast films, which have comparatively large crystals; slower, fine-grain emulsions are therefore preferable in applications where fine detail is desired or where enlargement of the negatives is necessary.

Resolving power

The fineness of detail that a film can resolve depends not only on its graininess, but also on the light-scattering properties of the emulsion and on the contrast with which the film reproduces fine detail. Fine-grain films with thin emulsions yield the highest resolving power.

2.6.3 Diaphragm and Shutter Settings

Regardless of the type of film used, proper camera settings are essential in obtaining acceptable pictures. The principal settings are the lens diaphragm and shutter speed.

In the lens diaphragm, a series of leaves increase or decrease the size of the opening to control the amount of light passing through the lens to the film. The diaphragm control ring is calibrated with a scale of so-called *f*-numbers or stop numbers in a series such as: 1.4, 2, 2.8, 4, 5.6, 8, 11, 16, 22, and 32. The *f*-numbers are inversely proportional to the amount of light admitted. In the above series, each setting admits twice as much light as the next higher *f*-number (thus giving twice as much exposure), and half as much light as the next lower value. Shutter speed settings on present-day cameras also follow a standard double-or-half sequence. Typical speeds are 1, ½, ¼, ⅛, 1/15, 1/30, 1/60, 1/125, 1/250, 1/500, and 1/1000 sec. The faster the shutter speed, the shorter the exposure time obtained.

The diaphragm and shutter control the amount of light reaching the film by adjusting the light intensity and the time during which it acts. Different aperture–shutter speed combinations can thus yield the same exposure. For example, diaphragm *f*/2.8 with 1/250 sec, *f*/4 with 1/125 sec, and *f*/5.6 with 1/60 sec, all yield the same exposure. It should be noted, however, that the combination chosen for these two settings is not independent of the conditions under which a picture is taken. For example, when photographing a scene where depth of focus is of interest, the *f*-stop should be selected as high as possible to give the lens a "pin-hole" characteristic. For a given film, this requirement limits the range of shutter speeds that yield adequate exposures. In other applications, the shutter speed is the essential consideration. An example with image processing implications is the problem of photographing a television screen. In this case, the shutter speed must be set below the refreshing rate of the TV set (1/30 sec per frame) or trace back information that is too fast for the human eye to see will be recorded on the film. Typically, ⅛ sec is adequate, although slower speeds are often used. Many of the images in this book, for example, were photographed at ¼ sec with Kodak Panatomic-X fine-grain film (ASA 32). The diaphragm settings were determined by using an exposure meter to measure the light intensity of each image.

2.7 CONCLUDING REMARKS

The material in this chapter is primarily background information for subsequent discussions. Our treatment of the human visual system, although brief, should give the reader a basic idea of the capabilities of the eye in perceiving pictorial information. Similarly, the image model developed in Section 2.2 is used in Chapter 4 as the basis for an image enhancement technique called *homomorphic filtering*.

The sampling ideas introduced in Section 2.3 are considered again in Section 3.3.9 after the necessary mathematical tools for a deeper analytical study of this problem are developed. Sampling and quantization considerations also play a central role in Chapter 6 in the context of image encoding applications, where the problem is one of compressing the large quantities of data that result from image digitization.

The material in Section 2.4 is basic to the understanding of numerous image processing techniques discussed in the following chapters. The imaging geometry

concepts developed in Section 2.5 play an important role in situations where three-dimensional scene information must be correlated with images acquired by a camera and subsequently processed by a computer.

REFERENCES

The material presented in Sections 2.1.1 and 2.1.2 is based primarily on the books by Cornsweet [1970] and by Graham [1965]. Additional reading for Section 2.1.3 may be found in Sheppard [1968]; Sheppard, Stratton, and Gazley [1969]; and Stevens [1951]. The image model presented in Section 2.2 has been investigated by Oppenheim, Schafer, and Stockham [1968] in connection with image enhancement applications. References for the illumination and reflectance values used in that section are Moon [1961] and the *IES Lighting Handbook* [1972]. Some of the material presented in Section 2.3 is based on the work of Huang [1965]. The papers by Scoville [1965] and by Gaven, Tavitian, and Harabedian [1970] are also of interest. Additional reading for the material in Section 2.4 may be found in Toriwaki et al. [1979] and in Rosenfeld and Kak [1982]. Section 2.5 is from the book by Fu, Gonzalez, and Lee [1987]. References for Section 2.6 are Mees [1966], Perrin [1960], Nelson [1971], and *Kodak Plates and Films for Scientific Photography* [1973].

PROBLEMS

2.1 Suppose that a flat area with center at (x_0, y_0) is illuminated by a light source with intensity distribution

$$i(x, y) = Ke^{-[(x-x_0)^2+(y-y_0)^2]}.$$

If the reflectance characteristic of the area is

$$r(x, y) = 10(x - x_0) + 10(y - y_0) + 20.$$

What is the value of K that would yield an image intensity of 100 at (x_0, y_0)?

2.2 Assume that the area in Problem 2.1 now has a constant reflectance of 1, and let $K = 255$. If the resulting image is digitized with n bits of intensity resolution, and the eye can detect an abrupt change of eight shades of intensity between adjacent pixels, what is the value of n that will cause visible false contouring?

2.3 Sketch what the image in Problem 2.2 would look like for $n = 2$.

2.4 A common measure of transmission for digital data is the *baud rate*, defined as the number of bits transmitted per second. Generally, transmission is accomplished in packets consisting of a start bit, a byte (8 bits) of information, and a stop bit. Using this approach, answer the following:

a) How many minutes would it take to transmit a 512 × 512 image with 256 gray levels at 300 baud?

b) What would the time be at 9600 baud?

c) Repeat (a) and (b) for a 1024 × 1024 image with 256 gray levels.

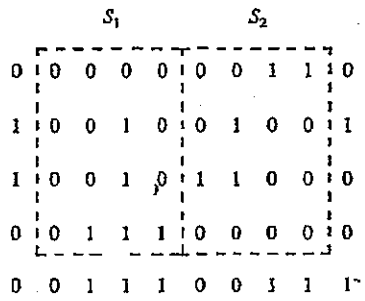
2.5 a) Show that the D_4 distance between two points p and q is equal to the shortest 4-path between these points.

b) Is this path unique?

2.6 Consider the image segment shown below.
a) Let $V = \{0, 1\}$ and compute the D_4, D_8, and D_m distances between p and q.
b) Repeat for $V = \{1, 2\}$.

3	1	2	1(q)
2	2	0	2
1	2	1	1
(p)1	0	1	2

2.7 Consider the two image subsets S_1 and S_2 shown below. Given that $V = \{1\}$, determine if S_1 and S_2 are (a) 4-connected, (b) 8-connected, (c) m-connected.



2.8 a) Give the transformation matrix used to rotate an *image* by 45° in the clockwise direction.
b) How would this transformation be used to achieve the desired image rotation?
c) Use the matrix obtained in (a) to rotate the image point $(x, y) = (1, 0)$.

2.9 Determine if the world point with coordinates $(1/2, 1/2, \sqrt{2}/2)$ is on the optical axis of a camera located at $(0, 0, \sqrt{2})$, panned 135°, and tilted 135°. Assume a 50-mm lens and let $r_1 = r_2 = r_3 = 0$.

2.10 Start with Eq. (2.5-41) and derive Eqs. (2.5-42) and (2.5-43).

2.11 Modify the ALU procedure given in Section 2.4.4 to replace each pixel in an image by the average of its 4-neighbors. Do not include the pixel itself in the computation of the average.

2.12 The 3 × 3 mask shown below is frequently used to compute the derivative in the x direction at each point in an image.

−1	−2	−1
0	0	0
1	2	1

Give an ALU procedure used to implement this operation.

CHAPTER 4

IMAGE ENHANCEMENT

> It makes all the difference whether one sees
> darkness through the light or brightness
> through the shadows.
> *David Lindsay*

The principal objective of enhancement techniques is to process a given image so that the result is more suitable than the original image for a specific application. The word "specific" is important because it establishes at the outset that the techniques discussed in this chapter are very much problem-oriented. Thus, for example, a method that is quite useful for enhancing x-ray images may not necessarily be the best approach for enhancing pictures of Mars transmitted by a space probe.

The approaches discussed in this chapter may be divided into two broad categories: frequency-domain methods and spatial-domain methods. Processing techniques in the first category are based on modifying the Fourier transform of an image. The spatial domain, on the other hand, refers to the image plane itself, and approaches in this category are based on direct manipulation of the pixels in an image.

The basic methodology underlying the material developed in this chapter is presented in Section 4.1. Section 4.2 deals with image enhancement by histogram-modification techniques. Sections 4.3 and 4.4 contain a number of approaches for image smoothing and sharpening, respectively. This discussion is followed in Section 4.5 by an enhancement technique based on the illumination-reflectance model introduced in Section 2.2. In Section 4.6 we develop a technique for generating small spatial masks from a filter function specified in the frequency domain. Finally, Section 4.7 contains an introduction to color fundamentals, as well as several applications of pseudo-color concepts to image enhancement.

4.1 BACKGROUND

The image-enhancement methods presented in this chapter are based on either *spatial-* or *frequency-domain* techniques. The purpose of this section is to develop the fundamental ideas underlying and relating these two approaches.

4.1.1 Spatial-Domain Methods

The term *spatial domain* refers to the aggregate of pixels composing an image, and spatial-domain methods are procedures that operate directly on these pixels. Image-processing functions in the spatial domain may be expressed as

$$g(x, y) = T[f(x, y)], \tag{4.1-1}$$

where $f(x, y)$ is the input image, $g(x, y)$ is the processed image, and T is an operator on f, defined over some neighborhood of (x, y). It is also possible to let T operate on a *set* of input images, such as performing the pixel-by-pixel sum of K images for noise reduction, as discussed in Section 4.3.4.

The principal approach used in defining a neighborhood about (x, y) is to use a square or rectangular subimage area centered at (x, y), as shown in Fig. 4.1. The center of the subimage is moved from pixel to pixel starting, say, at the top left corner, and applying the operator at each location (x, y) to yield the value of g at that location. Although other neighborhood shapes, such as a circle, are sometimes used, square arrays are by far the most predominant because of their ease of implementation.

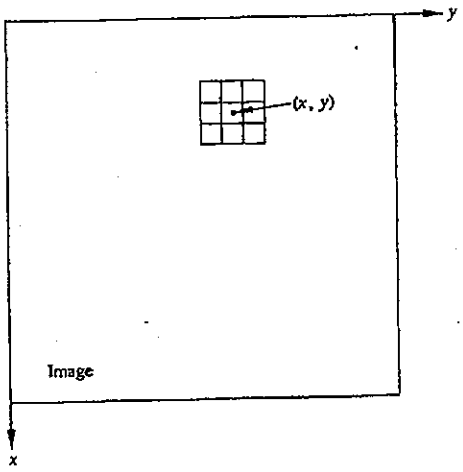


Figure 4.1 A 3 × 3 neighborhood about a point (x, y) in an image.

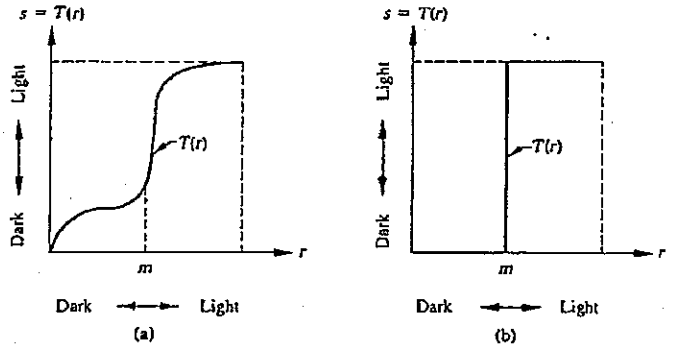


Figure 4.2 Gray-level transformation functions for contrast enhancement.

The simplest form of T is when the neighborhood is 1×1. In this case, g depends only on the value of f at (x, y) and T becomes a gray-level *transformation* (also called *mapping*) *function* of the form

$$s = T(r), \tag{4.1-2}$$

where, for simplicity in notation, we use r and s as variables that denote the gray level of $f(x, y)$ and $g(x, y)$ at any point (x, y). As an illustration, if $T(r)$ has the form shown in Fig. 4.2(a), the effect of this transformation is to produce an image of higher contrast than the original by darkening the levels below a value m and brightening the levels above m in the original pixel spectrum. In this technique, known as *contrast stretching*, the levels of r below m are compressed by the transformation function into a narrow range of s toward the dark end of the spectrum; the opposite effect takes place for values of r above m. In the limiting case shown in Fig. 4.2(b), $T(r)$ produces a two-level (i.e., binary) image. As shown in the following sections, some fairly simple, yet powerful, processing approaches can be formulated using gray-level transformations.

Larger neighborhoods allow a variety of processing functions that, as will be seen in subsequent chapters, go beyond just image enhancement. Regardless of the specific application, however, the general approach is to let the values of f in a predefined neighborhood of (x, y) determine the value of g at those coordinates. One of the principal approaches in this formulation is based on the use of so-called *masks* (also referred to as *templates*, *windows*, or *filters*). Basically, a mask is a small (e.g., 3×3) two-dimensional array, such as the one shown in Fig. 4.1, whose coefficients are chosen to detect a given property in an image. As an introduction to this concept, suppose that we have an image of constant intensity that contains widely isolated points whose intensities are different from the background. These points can be detected by using the mask shown in Fig. 4.3. The procedure is as

−1	−1	−1
−1	8	−1
−1	−1	−1

Figure 4.3 A mask for detecting isolated points different from a constant background.

follows: The center of the mask (labeled 8) is moved around the image, as indicated in Fig. 4.1. At each pixel position in the image, we multiply every pixel that is contained within the mask area by the corresponding mask coefficient; that is, the pixel in the center of the mask is multiplied by 8, while its 8-neighbors are multiplied by −1. The results of these nine multiplications are then summed. If all the pixels within the mask area have the same value (constant background), the sum will be zero. If, on the other hand, the center of the mask is located at one of the isolated points, the sum will be different from zero. If the isolated point is in an off-center position the sum will also be different from zero, but the magnitude of the response will be weaker. These weaker responses can be eliminated by comparing the sum against a threshold.

As shown in Fig. 4.4, if we let $w_1, w_2, \ldots, w_9$ represent mask coefficients and consider the 8-neighbors of (x, y), we may generalize the preceding discussion

w_1 $(x-1, y-1)$	w_2 $(x-1, y)$	w_3 $(x-1, y+1)$
w_4 $(x, y-1)$	w_5 (x, y)	w_6 $(x, y+1)$
w_7 $(x+1, y-1)$	w_8 $(x+1, y)$	w_9 $(x+1, y+1)$

Figure 4.4 A general 3 × 3 mask showing coefficients and corresponding image pixel locations.

as performing the following operation:

$$\begin{aligned} T[f(x, y)] = {} & w_1 f(x-1, y-1) + w_2 f(x-1, y) \\ & + w_3 f(x-1, y+1) + w_4 f(x, y-1) \\ & + w_5 f(x, y) + w_6 f(x, y+1) + w_7 f(x+1, y-1) \\ & + w_8 f(x+1, y) + w_9 f(x+1, y+1) \end{aligned} \tag{4.1-3}$$

on a 3×3 neighborhood of (x, y). Larger masks are formed in a similar manner.

It is noted in Eq. (4.1-3) that changing the coefficients changes the function of the mask. For instance, if we select $w_i = 1/9$, $i = 1, 2, \ldots 9$, and let $g(x, y) = T[f(x, y)]$, then the values of g at (x, y) will be the average gray level of the pixel at (x, y) and its 8-neighbors. As indicated above, masks have a number of uses besides image enhancement. Some of these uses include image restoration, object segmentation, and computing the skeleton of a binary region.

4.1.2 Frequency-Domain Methods

The foundation of frequency-domain techniques is the convolution theorem. Let $g(x, y)$ be an image formed by the convolution of an image $f(x, y)$ and a position-invariant operator $h(x, y)$,[†] that is,

$$g(x, y) = h(x, y) * f(x, y). \tag{4.1-4}$$

Then, from the convolution theorem (Section 3.3.8), we have that the following frequency-domain relation holds:

$$G(u, v) = H(u, v)\, F(u, v), \tag{4.1-5}$$

where G, H, and F are the Fourier transforms of g, h, and f, respectively. The transform $H(u, v)$ is sometimes called the *transfer function* of the process.

It is shown later in this chapter that numerous image-enhancement problems can be expressed in the form of Eq (4.1-5). In a typical image-enhancement application, $f(x, y)$ is given and the goal, after computation of $F(u, v)$, is to select $H(u, v)$ so that the desired image, given by,

$$g(x, y) = \mathfrak{F}^{-1}[H(u, v)F(u, v)] \tag{4.1-6}$$

exhibits some highlighted feature of $f(x, y)$. For instance, edges in $f(x, y)$ can be accentuated by using a function $H(u, v)$ that emphasizes the high-frequency components of $F(u, v)$.

It is worth noting that Eq. (4.1-4) describes a spatial process that is analogous to the use of the masks discussed in the previous section. In fact, the discrete convolution expression given in Eq. (3.3-35) is basically a mathematical representation of the mechanics involved in implementing the mask-shifting process explained in

[†] A position-invariant operator is one whose result depends only on the value of $f(x, y)$ at a given point in the image and not on the position of the point. Position invariance is an implicit requirement in the definition of the convolution integrals given in Eqs. (3.3-23) and (3.3-30).

Fig. 4.1. For this reason, the spatial masks discussed in the previous section are often called *convolution masks*. The key point to keep in mind, however, is that the convolution theorem requires that $H(u, v)$ and $h(x, y)$ be of the same size. Thus if $H(u, v)$ is, for example, of size 512×512, then Eq. (4.1-4) gives an equivalent spatial result only if $h(x, y)$ is of this size. As indicated in Section 3.3.8, a discrete convolution with large arrays is more efficiently carried out in the frequency domain via the fast Fourier transform (FFT). In Section 4.6 we develop a method for obtaining small spatial convolution masks that *approximate* a given $H(u, v)$ in a least-square-error sense and can thus be used for spatial processing. Finally, we point out that, as indicated in Section 3.3.8, discrete convolution is characterized by wraparound error unless the functions are assumed to be periodic with periods chosen according to Eqs. (3.3-33) and (3.3-34). In an image, wraparound error manifests itself as a distortion around the edges. In practice, however, this error is often not objectionable, even when the images are not extended using the procedure given in Section 3.3.8. The results given later in this chapter, for example, were obtained by direct FFT computations on the given images without extension.

Although it may already be obvious, let us emphasize before leaving this section that there is no general theory of image enhancement. When an image is processed for visual interpretation, the viewer is the ultimate judge of how well a particular method works. Visual evaluation of image quality is a highly subjective process, thus making the definition of a "good image" an elusive standard by which to compare algorithm performance. When the problem is one of processing images for machine perception, the evaluation task is somewhat easier. If, for example, one were dealing with a character-recognition application, the best image-processing method would be the one yielding the best machine-recognition results. It is noteworthy, however, that even in situations where a clear-cut criterion of performance can be imposed on the problem, one usually is still faced with a certain amount of trial and error before being able to settle on a particular image-processing approach.

4.2 IMAGE ENHANCEMENT BY HISTOGRAM-MODIFICATION TECHNIQUES

A histogram of gray-level content provides a global description of the appearance of an image. The methods discussed in this section achieve enhancement by modifying the histogram of a given image in a specified manner. The type and degree of enhancement obtained depends on the nature of the specified histogram.

4.2.1 Foundation

Let the variable r represent the gray level of the pixels in the image to be enhanced. For simplicity, it will be assumed in the following discussion that the pixel values have been normalized so that they lie in the range

$$0 \leq r \leq 1, \qquad (4.2\text{-}1)$$

with $r = 0$ representing black and $r = 1$ representing white in the gray scale.

For any r in the interval [0, 1], attention will be focused on transformations of the form

$$s = T(r), \tag{4.2-2}$$

which produce a level s for every pixel value r in the original image. It is assumed that the transformation function given in Eq. (4.2-2) satisfies the conditions:

(a) $T(r)$ is single-valued and monotonically increasing in the interval $0 \leq r \leq 1$, and

(b) $0 \leq T(r) \leq 1$ for $0 \leq r \leq 1$.

Condition (a) preserves the order from black to white in the gray scale, while condition (b) guarantees a mapping that is consistent with the allowed range of pixel values. A transformation function satisfying these conditions is illustrated in Fig. 4.5.

The inverse transformation from s back to r will be denoted by

$$r = T^{-1}(s) \qquad 0 \leq s \leq 1, \tag{4.2-3}$$

where it is assumed that $T^{-1}(s)$ also satisfies conditions (a) and (b) with respect to the variable s.

The gray levels in an image are random quantities in the interval [0, 1]. Assuming for a moment that they are continuous variables, the original and transformed gray levels can be characterized by their probability density functions $p_r(r)$ and $p_s(s)$, respectively. A great deal can be said about the general characteristics of an image

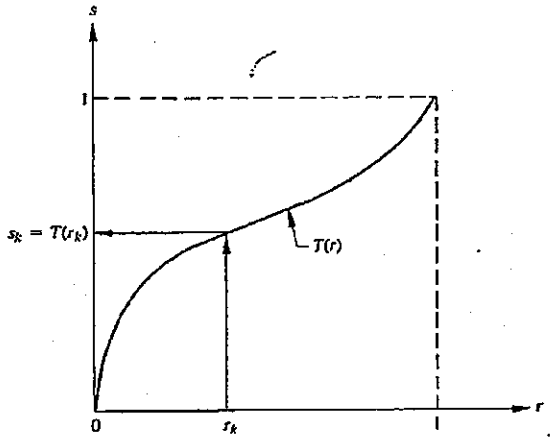


Figure 4.5 A gray-level transformation function.

from the density function of its gray levels. For example, an image whose gray levels have a density function like the one shown in Fig. 4.6(a) would have fairly dark characteristics since the majority of its levels are concentrated in the dark region of the gray scale. An image whose gray levels have a density function like the one shown in Fig. 4.6(b), on the other hand, would have predominant light tones since the majority of its pixels are light gray.

It follows from elementary probability theory that if $p_r(r)$ and $T(r)$ are known, and $T^{-1}(s)$ satisfies condition (a), then the probability density function of the transformed gray levels is given by the relation

$$p_s(s) = \left[p_r(r) \frac{dr}{ds} \right]_{r=T^{-1}(s)} \tag{4.2-4}$$

The following enhancement techniques are based on modifying the appearance of an image by controlling the probability density function of its gray levels via the transformation function $T(r)$.

4.2.2 Histogram Equalization

Consider the transformation function

$$s = T(r) = \int_0^r p_r(w)\, dw \qquad 0 \leq r \leq 1, \tag{4.2-5}$$

where w is a dummy variable of integration. The rightmost side of Eq. (4.2-5) is recognized as the cumulative distribution function (CDF) of r. The two conditions set forth in the previous section are satisfied by this transformation function since the CDF increases monotonically from 0 to 1 as a function of r.

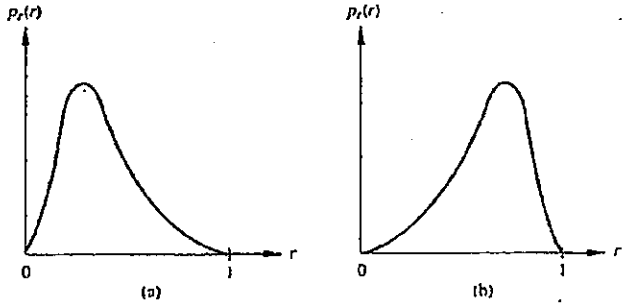


Figure 4.6 Gray-level probability density functions of (a) a "dark" image, and (b) a "light" image.

From Eq. (4.2-5) the derivative of s with respect to r is given by

$$\frac{ds}{dr} = p_r(r). \tag{4.2-6}$$

Substituting dr/ds into Eq. (4.2-4) yields

$$\begin{aligned} p_s(s) &= \left[p_r(r)\frac{1}{p_r(r)}\right]_{r=T^{-1}(s)} \\ &= [1]_{r=T^{-1}(s)} \\ &= 1 \qquad 0 \leq s \leq 1, \end{aligned} \tag{4.2-7}$$

which is a uniform density in the interval of definition of the transformed variable s. It is noted that this result is independent of the inverse transformation function. This is important because it is not always easy to obtain $T^{-1}(s)$ analytically.

The foregoing development indicates that using a transformation function equal to the cumulative distribution of r produces an image whose gray levels have a uniform density. In terms of enhancement, this implies an increase in the dynamic range of the pixels, which, as will be seen below, can have a considerable effect in the appearance of an image.

Example: Before proceeding with a discussion of discrete variables, let us consider a simple illustration of the use of Eqs. (4.2-4) and (4.2-5). Assume that the levels r have the probability density function shown in Fig. 4.7(a). In this case $p_r(r)$ is given by

$$p_r(r) = \begin{cases} -2r + 2 & 0 \leq r \leq 1 \\ 0 & \text{elsewhere.} \end{cases}$$

Substitution of this expression in Eq. (4.2-5) yields the transformation function

$$\begin{aligned} s = T(r) &= \int_0^r (-2w + 2)\, dw \\ &= -r^2 + 2r. \end{aligned}$$

Although we only need $T(r)$ for histogram equalization, it will be instructive to show that the resulting density $p_s(s)$ is in fact uniform since this requires that $T^{-1}(s)$ be obtained. In practice this step is not required because Eq. (4.2-7) is independent of the inverse transformation function. Solving for r in terms of s yields

$$r = T^{-1}(s) = 1 \pm \sqrt{1 - s}.$$

Since r lies in the interval [0, 1], only the solution

$$r = T^{-1}(s) = 1 - \sqrt{1 - s}$$

is valid.

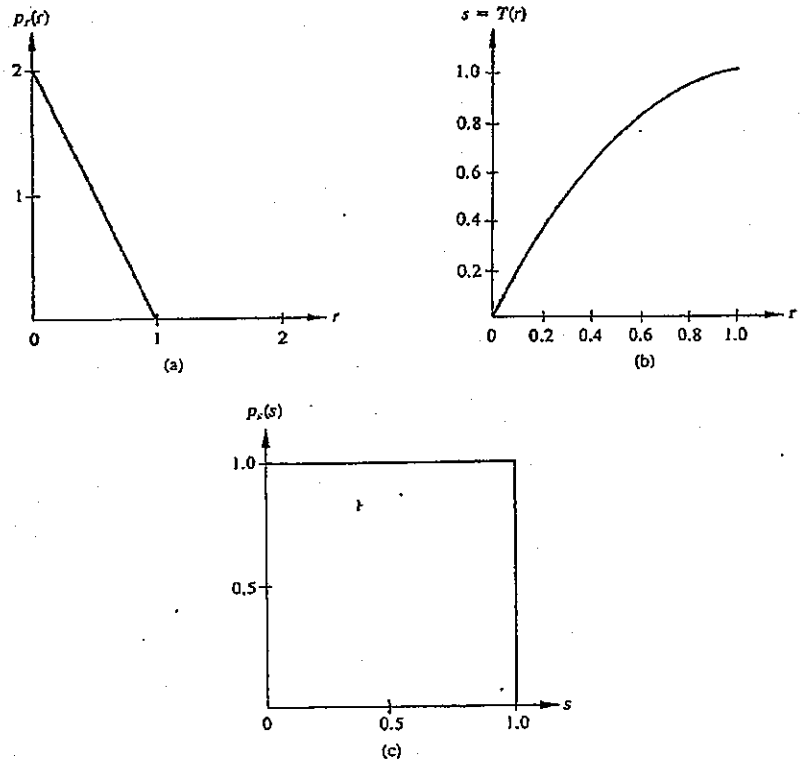


Figure 4.7 Illustration of the uniform-density transformation method. (a) Original probability density function. (b) Transformation function. (c) Resulting uniform density.

The probability density function of s is obtained by substituting the above results into Eq. (4.2-4):

$$\begin{aligned} p_s(s) &= \left[p_r(r) \frac{dr}{ds} \right]_{r=T^{-1}(s)} \\ &= \left[(-2r + 2) \frac{dr}{ds} \right]_{r=1-\sqrt{1-s}} \\ &= \left[(2\sqrt{1-s}) \frac{d}{ds} (1 - \sqrt{1-s}) \right] \\ &= 1 \qquad 0 \leq s \leq 1, \end{aligned}$$

which is a uniform density in the desired range. The transformation function $T(r)$ is shown in Fig. 4.7(b), and $p_s(s)$ is shown in Fig. 4.7(c). □

In order to be useful for digital image processing, the concepts developed above must be formulated in discrete form. For gray levels that assume discrete values, we deal with probabilities given by the relation

$$p_r(r_k) = \frac{n_k}{n} \qquad \begin{matrix} 0 \leq r_k \leq 1 \\ k = 0, 1, \ldots, L-1, \end{matrix} \tag{4.2-8}$$

where L is the number of levels, $p_r(r_k)$ is the probability of the kth gray level, n_k is the number of times this level appears in the image, and n is the total number of pixels in the image. A plot of $p_r(r_k)$ versus r_k is usually called a *histogram*, and the technique used for obtaining a uniform histogram is known as *histogram equalization* or *histogram linearization*.

The discrete form of Eq. (4.2-5) is given by the relation

$$\begin{aligned} s_k = T(r_k) &= \sum_{j=0}^{k} \frac{n_j}{n} \\ &= \sum_{j=0}^{k} p_r(r_j) \end{aligned} \qquad \begin{matrix} 0 \leq r_k \leq 1 \\ k = 0, 1, \ldots, L-1. \end{matrix} \tag{4.2-9}$$

The inverse transformation is denoted by

$$r_k = T^{-1}(s_k) \qquad 0 \leq s_k \leq 1,$$

where both $T(r_k)$ and $T^{-1}(s_k)$ are assumed to satisfy conditions (a) and (b) stated in the previous section. It is noted that the transformation function $T(r_k)$ can be computed directly from the image in question by using Eq. (4.2-9). Although the inverse function $T^{-1}(s_k)$ is not used in histogram equalization, it plays a central role in the method discussed in the next section.

Example: Suppose that a 64×64, 8-level image has the gray-level distribution shown in Table 4.1. The histogram of these gray levels is shown in Fig. 4.8(a).

Table 4.1

r_k	n_k	$p_r(r_k) = n_k/n$
$r_0 = 0$	790	0.19
$r_1 = 1/7$	1023	0.25
$r_2 = 2/7$	850	0.21
$r_3 = 3/7$	656	0.16
$r_4 = 4/7$	329	0.08
$r_5 = 5/7$	245	0.06
$r_6 = 6/7$	122	0.03
$r_7 = 1$	81	0.02

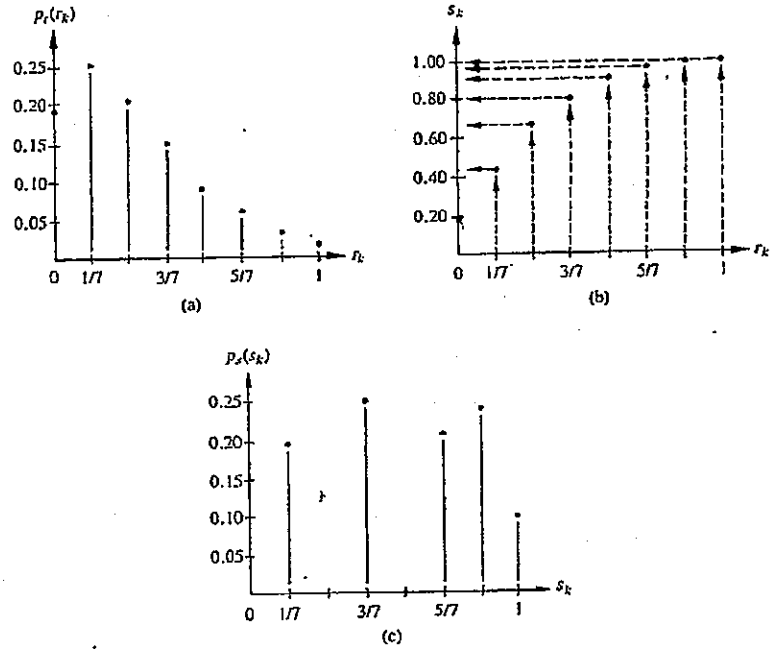


Figure 4.8 Illustration of the histogram-equalization method. (a) Original histogram. (b) Transformation function. (c) Equalized histogram.

The transformation function is obtained by using Eq. (4.2-9). For instance,

$$\begin{aligned} s_0 = T(r_0) &= \sum_{j=0}^{0} p_r(r_j) \\ &= p_r(r_0) \\ &= 0.19. \end{aligned}$$

Similarly,

$$\begin{aligned} s_1 = T(r_1) &= \sum_{j=0}^{1} p_r(r_j) \\ &= p_r(r_0) + p_r(r_1) \\ &= 0.44 \end{aligned}$$

and

$$s_2 = 0.65 \qquad s_5 = 0.95$$
$$s_3 = 0.81 \qquad s_6 = 0.98$$
$$s_4 = 0.89 \qquad s_7 = 1.00.$$

The transformation function has the form shown in Fig. 4.8(b).

Since only eight equally spaced levels are allowed in this case, each of the transformed values must be assigned to its closest valid level. Thus we have

$$s_0 \cong 1/7 \qquad s_4 \cong 6/7$$
$$s_1 \cong 3/7 \qquad s_5 \cong 1$$
$$s_2 \cong 5/7 \qquad s_6 \cong 1$$
$$s_3 \cong 6/7 \qquad s_7 \cong 1.$$

It is noted that there are only five distinct histogram-equalized gray levels. Redefining the notation to take this into account yields the levels

$$s_0 = 1/7 \qquad s_3 = 6/7$$
$$s_1 = 3/7 \qquad s_4 = 1.$$
$$s_2 = 5/7$$

Since $r_0 = 0$ was mapped to $s_0 = 1/7$, there are 790 transformed pixels with this new value. Also, there are 1023 pixels with value $s_1 = 3/7$ and 850 pixels with value $s_2 = 5/7$. However, since both levels r_3 and r_4 were mapped to $s_3 = 6/7$, there are now $656 + 329 = 985$ pixels with this new value. Similarly, there are $245 + 122 + 81 = 448$ pixels with value $s_4 = 1$. Dividing these numbers by $n = 4096$ yields the histogram shown in Fig. 4.8(c). Since a histogram is an approximation to a probability density function, perfectly flat results are seldom obtained when working with discrete levels. □

Example: As a practical illustration of histogram equalization consider the image shown in Fig. 4.9(a) containing a picture of a dollar, which, due to a heavy shadow, is barely visible. The narrow range of values occupied by the pixels of this image is evident in the histogram shown in Fig. 4.9(b). The equalized histogram is shown in Fig. 4.9(c) and the processed image in Fig. 4.9(d). While the equalized histogram is, as expected, not perfectly flat throughout the full range of gray levels, considerable improvement over the original image was achieved by the spreading effect of the histogram-equalization technique. Although the final result is not an ideal picture, one should keep in mind the poor quality of the original. □

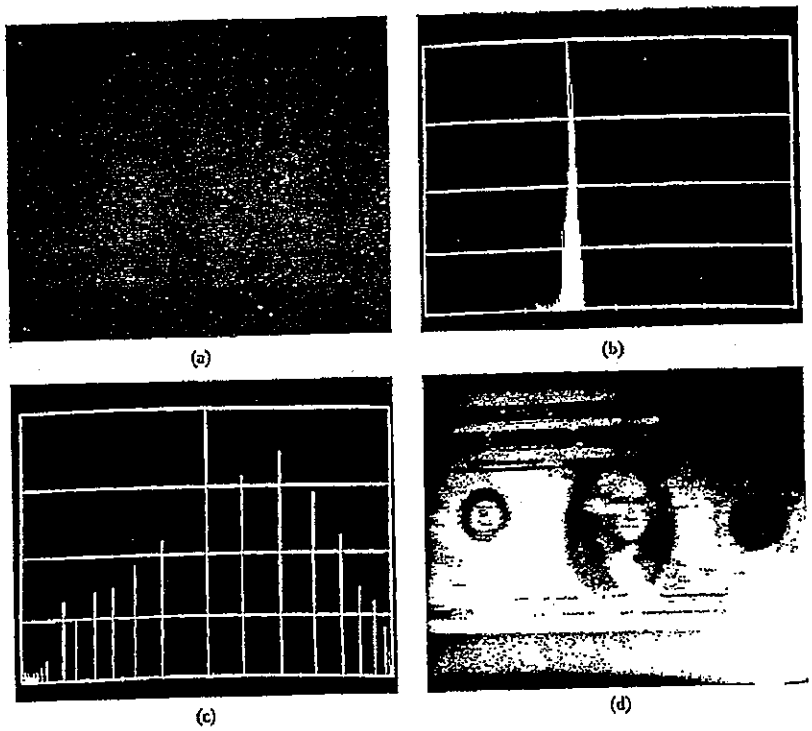


Figure 4.9 Illustration of the histogram-equalization approach. (a) Original image. (b) Original histogram. (c) Equalized histogram. (d) Enhanced image.

4.2.3 Direct Histogram Specification

Although the method discussed in the previous section is quite useful, it does not lend itself to interactive image-enhancement applications because the capabilities of this method are limited to the generation of only one result—an approximation to a uniform histogram.

It is sometimes desirable to be able to specify interactively particular histograms capable of highlighting certain gray-level ranges in an image. To see how this can be accomplished, let us return for a moment to continuous gray levels, and let $p_r(r)$ and $p_z(z)$ be the original and desired probability density functions, respectively. Suppose that a given image is first histogram equalized using Eq. (4.2-5); that is

$$s = T(r) = \int_0^r p_r(w)\, dw. \tag{4.2-10}$$

If the desired image were available, its levels could also be equalized by using the transformation function

$$v = G(z) = \int_0^z p_z(w)\,dw. \tag{4.2-11}$$

The inverse process, $z = G^{-1}(v)$, would then yield the desired levels back. This, of course, is a hypothetical formulation since the z levels are precisely what we are trying to obtain. It is noted, however, that $p_s(s)$ and $p_v(v)$ would be identical uniform densities since the final result of Eq. (4.2-5) is independent of the density inside the integral. Thus if instead of using v in the inverse process we use the uniform levels s obtained from the original image, the resulting levels, $z = G^{-1}(s)$, would have the desired probability density function. Assuming that $G^{-1}(s)$ is single-valued, the procedure can be summarized as follows:

(1) Equalize the levels of the original image using Eq. (4.2-5).
(2) Specify the desired density function and obtain the transformation function $G(z)$ using Eq. (4.2-11).
(3) Apply the inverse transformation function, $z = G^{-1}(s)$, to the levels obtained in Step (1).

This procedure yields a processed version of the original image, where the new gray levels are characterized by the specified density $p_z(z)$.

Although the method of histogram specification involves two transformation functions, $T(r)$ followed by $G^{-1}(s)$, it is a simple matter to combine both enhancement steps into one function that will yield the desired levels starting with the original pixels. From the above discussion, we have that

$$z = G^{-1}(s). \tag{4.2-12}$$

Substitution of Eq. (4.2-5) in Eq. (4.2-12) results in the combined transformation function

$$z = G^{-1}[T(r)], \tag{4.2-13}$$

which relates r to z. It is noted that, when $G^{-1}[T(r)] = T(r)$, this expression reduces to histogram equalization.

The implication of Eq. (4.2-13) is simply that an image need not be histogram-equalized explicitly. All that is required is that $T(r)$ be determined and combined with the inverse transformation function G^{-1}. The real problem in using the preceding method for continuous variables lies in obtaining the inverse function analytically. In the discrete case this problem is circumvented by the fact that the number of distinct gray levels is usually relatively small and it becomes feasible to calculate and store a mapping for each possible pixel value. The discrete formulation of the histogram-specification technique parallels Eqs. (4.2-8) and (4.2-9), as illustrated by the following example.

Example: Consider the 64 × 64, 8-level image values used in the second example given in Section 4.2.2. The histogram of this image is shown again in Fig. 4.10(a) for easy reference. It is desired to transform this histogram so that it will have the shape shown in Fig. 4.10(b). The values of the specified histogram are listed in Table 4.2.

The first step in the procedure is to obtain the histogram-equalization mappings. This was done in Section 4.2.2 with the results shown in Table 4.3. Next, we compute the transformation function using Eq. (4.2-9):

$$v_k = G(z_k) = \sum_{j=0}^{k} p_z(z_j).$$

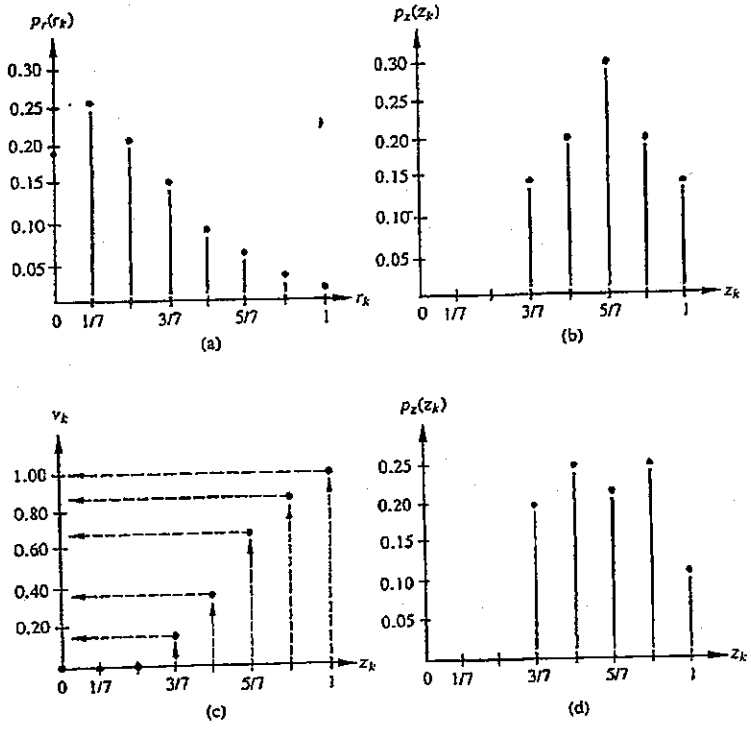


Figure 4.10 Illustration of the histogram-specification method. (a) Original histogram. (b) Specified histogram. (c) Transformation function. (d) Resulting histogram.

Table 4.2

z_k	$p_z(z_k)$
$z_0 = 0$	0.00
$z_1 = 1/7$	0.00
$z_2 = 2/7$	0.00
$z_3 = 3/7$	0.15
$z_4 = 4/7$	0.20
$z_5 = 5/7$	0.30
$z_6 = 6/7$	0.20
$z_7 = 1$	0.15

This yields the values

$$v_0 = G(z_0) = 0.00 \qquad v_4 = G(z_4) = 0.35$$
$$v_1 = G(z_1) = 0.00 \qquad v_5 = G(z_5) = 0.65$$
$$v_2 = G(z_2) = 0.00 \qquad v_6 = G(z_6) = 0.85$$
$$v_3 = G(z_3) = 0.15 \qquad v_7 = G(z_7) = 1.00.$$

The transformation function is shown in Fig. 4.10(c).

To obtain the z levels we apply the inverse of the G transformation obtained above to the histogram-equalized levels s_k. Since we are dealing with discrete values, an approximation must usually be made in the inverse mapping. For example, the closest match to $s_0 = 1/7 \approx 0.14$ is $G(z_3) = 0.15$ or, using the inverse, $G^{-1}(0.15) = z_3$. Thus s_0 is mapped to the level z_3. Using this procedure yields the following mappings:

$$s_0 = 1/7 \rightarrow z_3 = 3/7 \qquad s_3 = 6/7 \rightarrow z_6 = 6/7$$
$$s_1 = 3/7 \rightarrow z_4 = 4/7 \qquad s_4 = 1 \rightarrow z_7 = 1.$$
$$s_2 = 5/7 \rightarrow z_5 = 5/7$$

Table 4.3

$r_j \rightarrow s_k$	n_k	$p_s(s_k)$
$r_0 \rightarrow s_0 = 1/7$	790	0.19
$r_1 \rightarrow s_1 = 3/7$	1023	0.25
$r_2 \rightarrow s_2 = 5/7$	850	0.21
$r_3, r_4 \rightarrow s_3 = 6/7$	985	0.24
$r_5, r_6, r_7 \rightarrow s_4 = 1$	448	0.11

Table 4.4

z_k	n_k	$p_z(z_k)$
$z_0 = 0$	0	0.00
$z_1 = 1/7$	0	0.00
$z_2 = 2/7$	0	0.00
$z_3 = 3/7$	790	0.19
$z_4 = 4/7$	1023	0.25
$z_5 = 5/7$	850	0.21
$z_6 = 6/7$	985	0.24
$z_7 = 1$	448	0.11

As indicated in Eq. (4.2-13), these results can be combined with those of histogram equalization to yield the following direct mappings:

$$r_0 = 0 \rightarrow z_3 = 3/7 \qquad r_4 = 4/7 \rightarrow z_6 = 6/7$$
$$r_1 = 1/7 \rightarrow z_4 = 4/7 \qquad r_5 = 5/7 \rightarrow z_7 = 1$$
$$r_2 = 2/7 \rightarrow z_5 = 5/7 \qquad r_6 = 6/7 \rightarrow z_7 = 1$$
$$r_3 = 3/7 \rightarrow z_6 = 6/7 \qquad r_7 = 1 \rightarrow z_7 = 1.$$

Redistributing the pixels according to these mappings and dividing by $n = 4096$ results in the histogram shown in Fig. 4.10(d). The values are listed in Table 4.4.

Note that, although each of the specified levels was filled, the resulting histogram is not particularly close to the desired shape. As in the case of histogram equalization, this error is due to the fact that the transformation is guaranteed to yield exact results only in the continuous case. As the number of levels decreases, the error between the specified and resulting histograms tends to increase. As will be seen below, however, very useful enhancement results can be obtained even with an approximation to a desired histogram. □

In practice, the inverse transformation from s to z is often not single-valued. This situation arises when there are unfilled levels in the specified histogram (which makes the CDF remain constant over the unfilled intervals), or in the process of rounding off $G^{-1}(s)$ to the nearest allowable level, as was done in the above example. Generally, the easiest solution to this problem is to assign the levels in such a way as to match the given histogram as closely as possible.

The principal difficulty in applying the histogram specification method to image enhancement lies in being able to construct a meaningful histogram. Two solutions to this problem are as follows. The first is to specify a particular probability density function (i.e., Gaussian, Rayleigh, log-normal, etc.) and then form a histogram by

digitizing the given function. The second approach consists of specifying a histogram shape by means of a graphic device (like an interactive screen or drawing tablet) whose output is fed into the processor executing the histogram specification algorithm.

Example: As a practical illustration of the direct histogram-specification approach, consider Fig. 4.11(a), which shows a semi-dark room viewed from a doorway. Figure 4.11(b) shows the histogram-equalized image, and Fig. 4.11(c) is the result of interactive histogram specification. The histograms are shown in Fig. 4.11(d), which includes, from bottom to top, the original, equalized, specified, and resulting histograms.

Note that histogram equalization produced an image whose contrast was somewhat high, while the result shown in Fig. 4.11(c) has a much more balanced appearance.

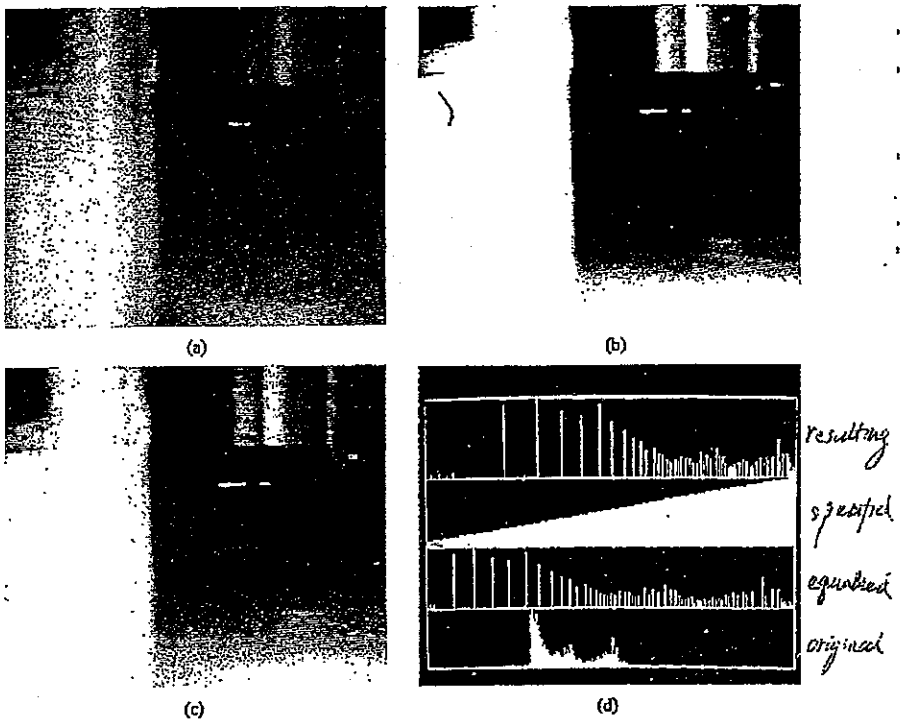


Figure 4.11 Illustration of the histogram-specification method. (a) Original image. (b) Histogram-equalized image. (c) Image enhanced by histogram specification. (d) Histograms.

Because of its flexibility, the direct-specification method can often yield results that are superior to histogram equalization. □

4.2.4 Local Enhancement

The methods discussed in the previous two sections are global, in the sense that pixels are modified by a transformation function based on the gray-level distribution over an entire image. While this global approach is suitable for overall enhancement, it is often necessary to enhance details over small areas. Since the number of pixels in these areas may have negligible influence on the computation of a global transformation, the use of this type of transformation will not necessarily guarantee the desired local enhancement. The solution is to devise transformation functions that are based on the gray-level distribution, or other properties, in the neighborhood of every pixel in a given image.

The histogram-processing techniques developed in the last two sections are easily adaptable to local enhancement. The procedure is to define an $n \times m$ neighborhood and move the center of this area from pixel to pixel. At each location, we compute the histogram of the $n \times m$ points in the neighborhood and obtain either a histogram-equalization or histogram-specification transformation function. This function is finally used to map the level of the pixel centered in the neighborhood. The center of the $n \times m$ region is then moved to an adjacent pixel location and the procedure is repeated. Since only one new row or column of the neighborhood changes during a pixel-to-pixel translation of the region, it is possible to update the histogram obtained in the previous location with the new data introduced at each motion step. This approach has obvious advantages over repeatedly computing the histogram over all $n \times m$ pixels every time the region is moved one pixel location. Another approach often used to reduce computation is to employ non-overlapping regions, but this will usually produce an undesirable checkerboard effect.

Example: An illustration of local histogram equalization with the neighborhood moved from pixel to pixel is shown in Fig. 4.12. Part (a) of this figure shows an image that has been slightly blurred to reduce its noise content (see Section 4.3.1). Figure 4.12(b) shows the result of global histogram equalization. As is often the case when this technique is applied to smooth, noisy areas, Fig. 4.12(b) shows considerable enhancement of the noise, with a slight increase in contrast. Note, however, that no new structural details were brought out by this method. On the other hand, local histogram equalization using a neighborhood of size 7×7 revealed the presence of small squares inside the larger dark squares. The small squares were too close in gray level and their size too small to influence global histogram equalization in any significant way. It is also of interest to note the finer noise texture in Fig. 4.12(c), a result of local processing in relatively small neighborhoods. □

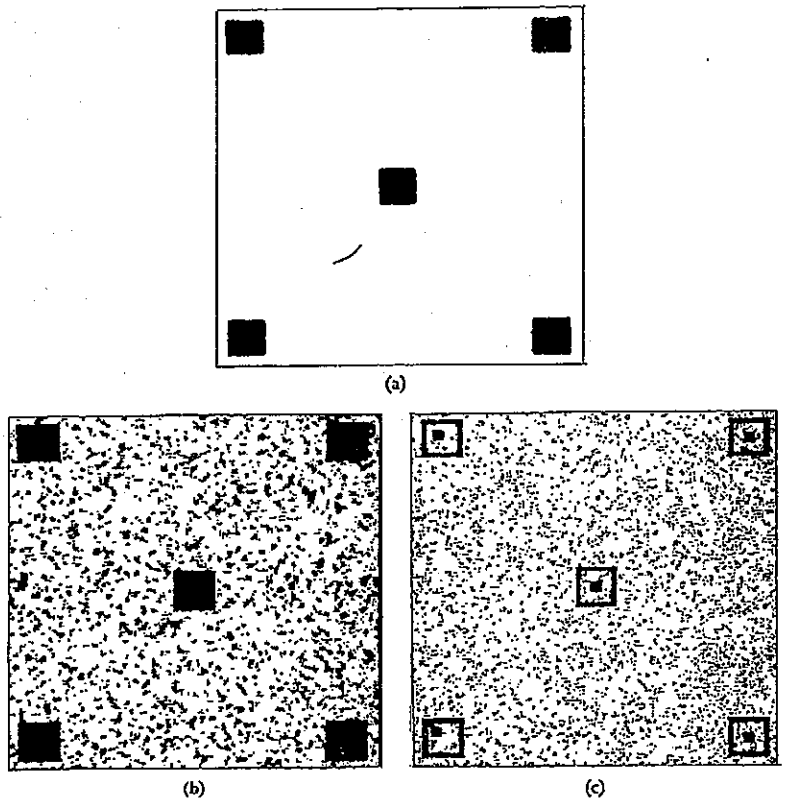


Figure 4.12 (a) Original image. (b) Result of global histogram equalization. (c) Result of local histogram equalization using a 7×7 neighborhood about each pixel. (From Fu, Gonzalez, and Lee [1987].)

Before leaving this section we point out that, instead of using histograms, one could base local enhancement on other properties of the pixel intensities in a neighborhood. The intensity mean and variance (or standard deviation) are two such properties frequently used because of their relevance to the appearance of an image. That is, the mean is a measure of average brightness and the variance is a measure of contrast.

A typical local transformation based on these concepts maps the intensity of an input image $f(x, y)$ into a new image $g(x, y)$ by performing the following transforma-

tion at each pixel location (x, y):

$$g(x, y) = A(x, y) \cdot [f(x, y) - m(x, y)] + m(x, y), \tag{4.2-14}$$

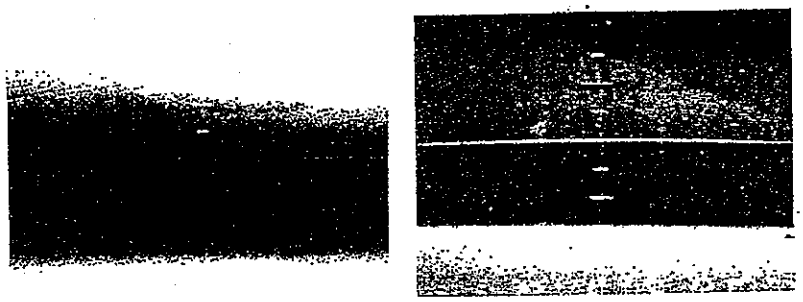
where

$$A(x, y) = k\frac{M}{\sigma(x, y)} \qquad 0 < k < 1. \tag{4.2-15}$$

In this formulation $m(x, y)$ and $\sigma(x, y)$ are the gray-level mean and standard deviation computed in a neighborhood centered at (x, y), M is the global mean of $f(x, y)$, and k is a constant in the range indicated above.

It is important to note that A, m, and σ are variable quantities that depend on a predefined neighborhood of (x, y). Application of the local gain factor $A(x, y)$ to the difference between $f(x, y)$ and the local mean amplifies local variations. Since $A(x, y)$ is inversely proportional to the standard deviation of the intensity, areas with low contrast receive larger gain. The mean is added back in Eq. (4.2-14) to restore the average intensity level of the image in the local region. In practice, it is often desirable to add back a fraction of the local mean and to restrict the variations of $A(x, y)$ between two limits (A_{min}, A_{max}) in order to balance large excursions of intensity in isolated regions.

Example: The preceding enhancement approach has been implemented in hardware by Narendra and Fitch [1981], and has the capability of processing images in real time (i.e., at 30 image frames/sec). An example of the capabilities of the technique using a local region on the order of 15 × 15 pixels is shown in Fig. 4.13. Note the enhancement of detail at the boundary between two regions of different overall gray levels and the rendition of gray-level details in each of the regions. □



(a) (b)

Figure 4.13 Images before and after local enhancement. (From Narendra and Fitch [1981].)

4.3 IMAGE SMOOTHING

Smoothing operations are used primarily for diminishing spurious effects that may be present in a digital image as a result of a poor sampling system or transmission channel. In this section we consider smoothing techniques in both the spatial and frequency domains.

4.3.1 Neighborhood Averaging

Neighborhood averaging is a straightforward spatial-domain technique for image smoothing. Given an $N \times N$ image $f(x, y)$, the procedure is to generate a smoothed image $g(x, y)$ whose gray level at every point (x, y) is obtained by averaging the gray-level values of the pixels of f contained in a predefined neighborhood of (x, y). In other words, the smoothed image is obtained by using the relation

$$g(x, y) = \frac{1}{M} \sum_{(n, m) \in S} f(n, m) \tag{4.3-1}$$

for $x, y = 0, 1, \ldots, N-1$. S is the set of coordinates of points in the neighborhood of the point (x, y), including (x, y) itself, and M is the total number of points in the neighborhood. If a 3×3 neighborhood is used, we note by comparing Eqs. (4.3-1) and (4.1-3) that the former equation is a special case of the latter with $w_i = 1/9$. Of course, we are not limited to square neighborhoods in Eq. (4.3-1), but these are by far the neighborhoods used most frequently due to ease of implementation.

Example: Figure 4.14 illustrates the smoothing effect produced by neighborhood averaging. Figure 4.14(a) is a simple four-level image corrupted by noise. Figures 4.14(b) through (f) are the results of processing the noisy image with neighborhoods of size $n \times n$, with $n = 3, 5, 9, 15$, and 31, respectively. It is noted that the degree of blurring is strongly proportional to the size of the neighborhood used. □

For a given neighborhood, the blurring effect produced by neighborhood averaging can be reduced by using a thresholding procedure; that is, instead of using Eq. (4.3-1), we form $g(x, y)$ according to the following criterion:

$$g(x, y) = \begin{cases} \dfrac{1}{M} \displaystyle\sum_{(m, n) \in S} f(m, n) & \text{if } \left| f(x, y) - \dfrac{1}{M} \sum_{(m, n) \in S} f(m, n) \right| < T \\ f(x, y) & \text{otherwise,} \end{cases} \tag{4.3-2}$$

where T is a specified nonnegative threshold. The motivation behind this approach is to reduce blurring by leaving unchanged regions of an image with large (compared to T) variations in gray level. Generally, we expect these variations to correspond to edges, so using Eq. (4.3-2) will reduce the amount of edge blurring. Other regions of the image are processed as before.

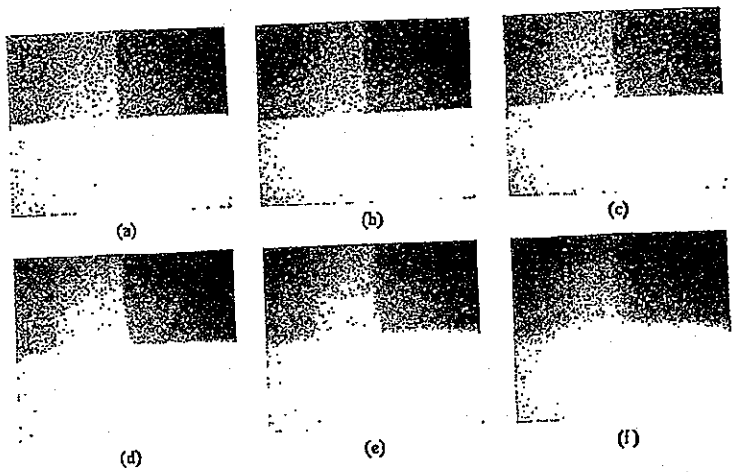


Figure 4.14 Example of neighborhood averaging. (a) Noisy image. (b) through (f), image processed using Eq. (4.3-1) with neighborhoods of size $n \times n$ with $n =$ 3, 5, 9, 15, and 31, respectively.

Example: Figure 4.15 shows the result of processing the image of Fig. 4.14(a) with Eq. (4.3-2). A 9×9 neighborhood and $T = 10$ were used in this case. We note by comparing this result with Fig. 4.14(d) that, although equivalent smoothing was achieved by use of a threshold, the boundaries between the four squares are much sharper in Fig. 4.15. □

4.3.2 Median Filtering

One of the principal difficulties of the method discussed in the previous section is that it blurs edges and other sharp details. Although this problem can be somewhat circumvented by using a threshold, choosing the threshold value generally involves considerable trial and error. An alternative approach is to use *median filters*, in which we replace the gray level of each pixel by the median of the gray levels in a neighborhood of that pixel, instead of by the average. This method is particularly effective when the noise pattern consists of strong, spikelike components, and where the characteristic to be preserved is edge sharpness.

Recall that the median m of a set of values is such that half of the values in the set are less than m and half are greater than m. In order to perform median filtering in a neighborhood of a pixel we first sort the values of the pixel and its neighbors, determine the median, and assign this value to the pixel. For example, in a 3×3 neighborhood the median is the 5th largest value, in a 5×5 neighborhood the 13th largest value, and so on. When several values in a neighborhood are the

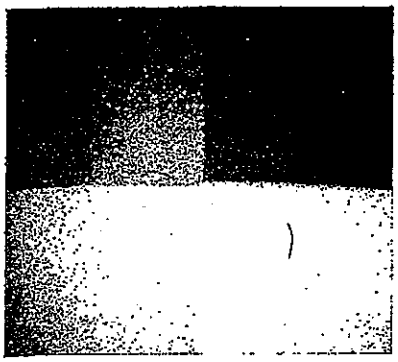


Figure 4.15 Smoothed image obtained by processing Fig. 4.14(a) with Eq. (4.3-2), using a 9×9 neighborhood and $T = 10$. Compare with Fig. 4.14(d).

same, we group all equal values as follows: Suppose that a 3×3 neighborhood has values (10, 20, 20, 20, 15, 20, 20, 25, 100). These values are sorted as (10, 15, 20, 20, 20, 20, 20, 25, 100), which results in a median of 20. A little thought will reveal that the principal function of median filtering is to force points with very distinct intensities to be more like their neighbors, thus actually eliminating intensity spikes that appear isolated in the area of the filter mask.

Example: Figure 4.16(a) shows an original image, and Fig. 4.16(b) shows the same image but with approximately 20% of the pixels corrupted by "impulse noise." The result of neighborhood averaging over a 5×5 area in shown in Fig. 4.16(c) and the result of a 5×5 median filter is shown in Fig. 4.16(d). The superiority of the median filter over neighborhood averaging needs no explanation. The bright dots remaining in Fig. 4.16(d) resulted from a large concentration of noise at those points, thus biasing the median calculation. Two or more passes with a median filter would eliminate these points. □

4.3.3 Lowpass Filtering

Edges and other sharp transitions (such as noise) in the gray levels of an image contribute heavily to the high-frequency content of its Fourier transform. It therefore follows that blurring can be achieved via the frequency domain by attenuating a specified range of high-frequency components in the transform of a given image.

From Eq. (4.1-5) we have the relation

$$G(u, v) = H(u, v)F(u, v), \tag{4.3-3}$$

where $F(u, v)$ is the transform of the image we wish to smooth. The problem is to select a function $H(u, v)$ that yields $G(u, v)$ by attenuating the high-frequency compo-

Figure 4.16 (a) Original image. (b) Image corrupted by impulse noise. (c) Result of 5 × 5 neighborhood averaging. (d) Result of 5 × 5 median filtering. (Courtesy of Martin Connor, Texas Instruments, Inc., Lewisville, Tex.)

nents of $F(u, v)$. The inverse transform of $G(u, v)$ will then yield the desired smoothed image $g(x, y)$. Since high-frequency components are "filtered out," and information in the low-frequency range is "passed" without attenuation, this method is commonly referred to as *lowpass filtering*. The function $H(u, v)$ is referred to in this context as a *filter transfer function*. Several low-pass-filtering approaches are discussed in the following paragraphs. In all cases, the filters are functions that affect corresponding real and imaginary components of the Fourier transform in exactly the same manner.

Such filters are referred to as *zero-phase-shift filters* because they do not alter the phase of the transform.

Ideal filter

A two-dimensional ideal lowpass filter (ILPF) is one whose transfer function satisfies the relation

$$H(u, v) = \begin{cases} 1 & \text{if } D(u, v) \leq D_0 \\ 0 & \text{if } D(u, v) > D_0, \end{cases} \tag{4.3-4}$$

where D_0 is a specified nonnegative quantity, and $D(u, v)$ is the distance from point (u, v) to the origin of the frequency plane; that is,

$$D(u, v) = (u^2 + v^2)^{1/2}. \tag{4.3-5}$$

A three-dimensional perspective plot of $H(u, v)$ versus u and v is shown in Fig. 4.17(a). The name *ideal* filter arises from the fact that all frequencies inside a circle of radius D_0 are passed with no attenuation, while all frequencies outside this circle are completely attenuated.

The lowpass filters considered in this chapter are radially symmetric about the origin. For this type of filter it suffices to specify a cross section extending as a function of distance from the origin along a radial line, as shown in Fig. 4.17(b). The complete filter transfer function can then be generated by rotating the cross section 360° about the origin. It should also be noted that specification of radially symmetric filters centered on the $N \times N$ frequency square is based on the assumption that the origin of the Fourier transform has been centered on the square, as discussed in Section 3.3.2.

For an ideal lowpass-filter cross section, the point of transition between $H(u, v) = 1$ and $H(u, v) = 0$ is often called the *cut-off frequency*. In the case of Fig. 4.17(b), for example, the cut-off frequency is D_0. As the cross section is rotated about the origin, the point D_0 traces a circle and we obtain a locus of cut-off frequencies

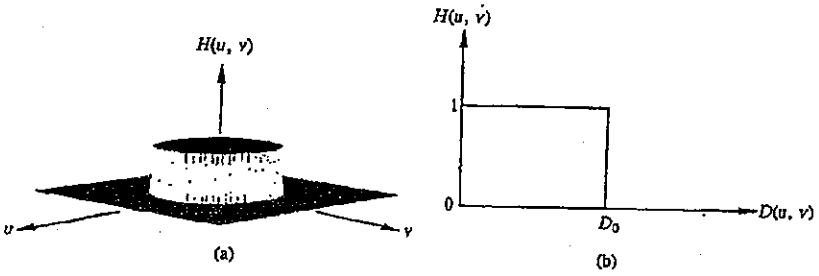


Figure 4.17 (a) Perspective plot of an ideal lowpass-filter transfer function. (b) Filter cross section.

all of which are a distance D_0 from the origin. As will be seen below, the cut-off–frequency concept is quite useful in specifying the characteristics of a given filter, and it also serves as a common base for comparing the behavior of different types of filters.

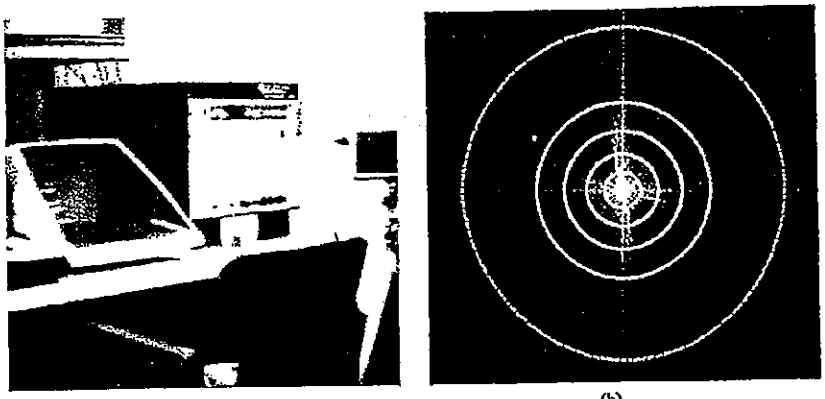
The sharp cut-off frequencies of an ideal lowpass filter cannot be realized with electronic components, although they can certainly be simulated in a computer. The effects of using these "nonphysical" filters on a digital image are discussed after the following example.

Example: The 256 × 256 image shown in Fig. 4.18(a) will be used to illustrate all the lowpass filters discussed in this section. The performance of these filters will be compared by using the same cut-off–frequency loci. One way to establish a set of standard loci is to compute circles that enclose various amounts of the total signal power P_T. This quantity is obtained by summing the power at each point (u, v) for $u, v = 0, 1, \ldots, N-1$; that is,

$$P_T = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} P(u, v),$$

where $P(u, v)$ is given by Eq. (3.1-13). Assuming that the transform has been centered, a circle of radius r with origin at the center of the frequency square encloses β percent of the power, where

$$\beta = 100\left[\sum_u \sum_v P(u, v)/P_T\right]$$



(a) (b)

Figure 4.18 (a) A 256 × 256 image, and (b) its Fourier spectrum. The superimposed circles, which have radii equal to 5, 11, 22, 36, 53, and 98, enclose, respectively, 90, 95, 98, 99, 99.5, and 99.9 percent of the image power.

and the summation is taken over values of (u, v) which lie inside the circle or on its boundary.

Figure 4.18(b) shows the Fourier transform of Fig. 4.18(a). The superimposed circles, which have radii of 5, 11, 22, 36, 53, and 98, enclose β percent of the power for $\beta = 90$, 95, 98, 99, 99.5, and 99.9, respectively. It is noted that the power spectrum falls off rather rapidly, with 90% of the total power being enclosed by the relatively small radius of 5. Since we are dealing with a 256×256 image and the Fourier transform has been centered, a circle of radius $(\sqrt{2})(128)$ would enclose 100% of the power.

The results of applying ideal lowpass filters with cut-off frequencies at the above radii are shown in Fig. 4.19. Part (a) of this figure is, for all practical purposes, useless. The severe blurring in this image is a clear indication that most of the edge information in the picture is contained within the 10% power removed by the filter. As the filter radius was increased the degree of blurring was, of course, decreased. It is interesting to note, however, that all the filtered images are characterized by considerable "ringing." This phenomenon, which is explained below, is visible even in Fig. 4.19(f), where only 0.1% of the power was removed. □

The blurring and ringing properties of the ILPF can be easily explained by resorting to the convolution theorem. Since the Fourier transforms of the original and blurred images are related in the frequency domain by the equation

$$G(u, v) = H(u, v)F(u, v),$$

it follows from this theorem that the following expression holds in the spatial domain:

$$g(x, y) = h(x, y) * f(x, y),$$

where $h(x, y)$ is the inverse Fourier transform of the filter transfer function $H(u, v)$.

The key to understanding blurring as a convolution process in the spatial domain lies in the nature of $h(x, y)$. For an ILPF, $h(x, y)$ has the general form shown in Fig. 4.20(a).† Suppose that $f(x, y)$ is a simple image composed of two bright pixels on a black background, as shown in Fig. 4.20(b). We may view the two bright points as approximations of two impulses whose strength depends on the brightness of these points. Then, the convolution of $h(x, y)$ and $f(x, y)$ is simply a process of "copying" $h(x, y)$ at the location of each impulse, as explained in Section 3.3.8. The result of this operation, which is shown in Fig. 4.20(c), explains how the two original points are blurred as a consequence of convolving $f(x, y)$ with the blurring function $h(x, y)$. These concepts are extended to more-complex images by considering each pixel as an impulse with a strength proportional to the gray level of the pixel.

† The reader can verify this by taking the inverse Fourier transform of Eq. (4.3-4).

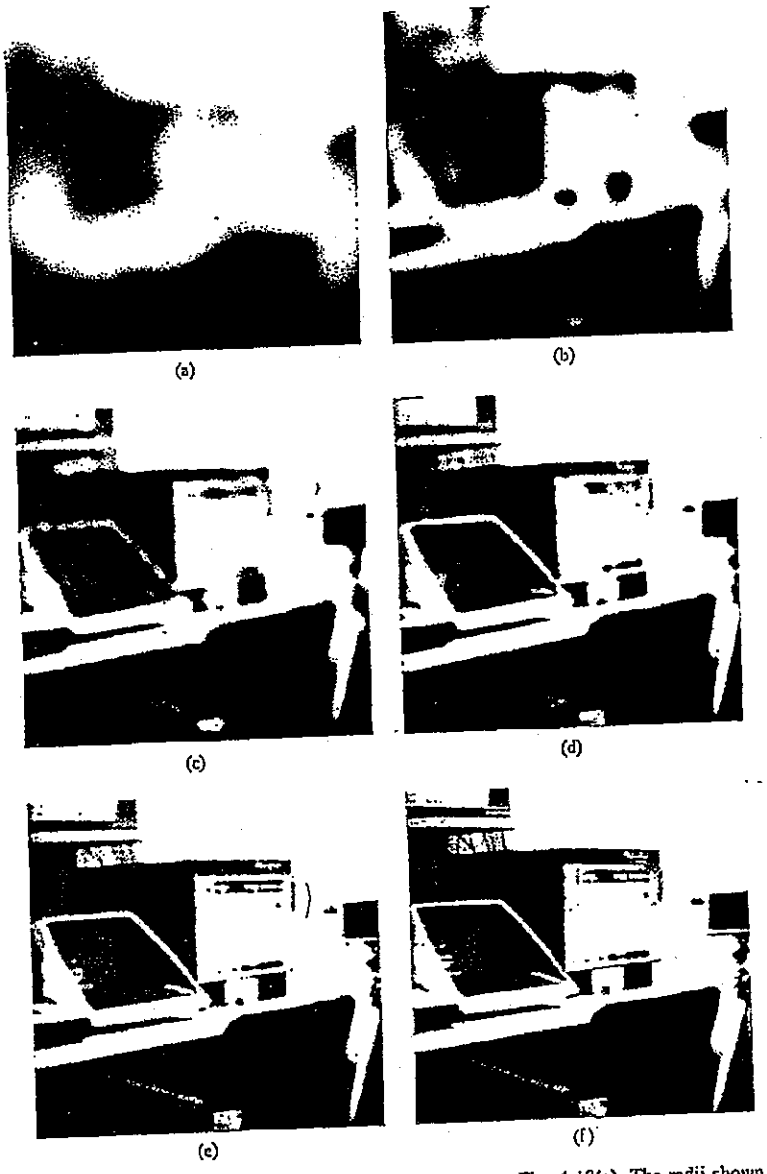
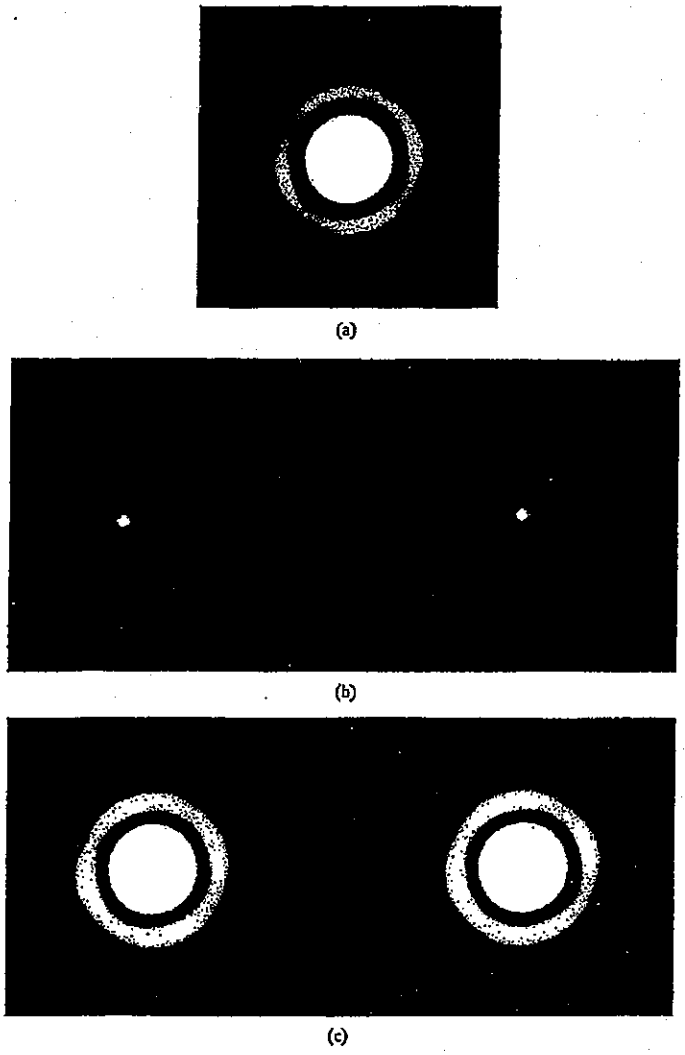


Figure 4.19 Results of applying ideal lowpass filters to Fig. 4.18(a). The radii shown in Fig. 4.18(b) were used.



(a)

(b)

(c)

Figure 4.20 Illustration of the blurring process in the spatial domain. (a) Blurring function $h(x, y)$ for an ideal lowpass filter. (b) A simple image composed of two bright dots. (c) Convolution of $h(x, y)$ and $f(x, y)$.

The shape of $h(x, y)$ depends on the radius of the filter function in the frequency domain. By computing the inverse transform of $H(u, v)$ for an ILPF, it can be shown that the radii of the concentric rings in $h(x, y)$ are inversely proportional to the value of D_0 in Eq. (4.3-4). Thus severe filtering in the frequency domain (i.e., choice of a small D_0) produces a relatively small number of broad rings in the $N \times N$ region of $h(x, y)$ and, consequently, pronounced blurring in $g(x, y)$. As D_0 increases, the number of rings in a given region increases, thus producing more finely spaced rings and less blurring. This effect can be observed by comparing Figs. 4.19(d) and (e). If D_0 is outside the $N \times N$ domain of definition of $F(u, v)$, $h(x, y)$ becomes unity in its corresponding $N \times N$ spatial region and the convolution of $h(x, y)$ and $f(x, y)$ is simply $f(x, y)$. This situation, of course, corresponds to no filtering at all. The spatial domain effect of the filters discussed below can be explained in a similar manner as for the ideal filter.

Butterworth filter

The transfer function of the Butterworth lowpass filter (BLPF) of order n and with cut-off frequency locus at a distance D_0 from the origin is defined by the relation

$$H(u, v) = \frac{1}{1 + [D(u, v)/D_0]^{2n}} \tag{4.3-6}$$

where $D(u, v)$ is given by Eq. (4.3-5). A perspective plot and cross section of the BLPF function are shown in Fig. 4.21.

Unlike the ILPF, the BLPF transfer function does not have a sharp discontinuity that establishes a clear cut-off between passed and filtered frequencies. For filters with smooth transfer functions it is customary to define a cut-off frequency locus at points for which $H(u, v)$ is down to a certain fraction of its maximum value. In the case of Eq. (4.3-6) we see that $H(u, v) = 0.5$ (down 50% from its maximum value of one) when $D(u, v) = D_0$. Another value commonly used is $1/\sqrt{2}$ of the

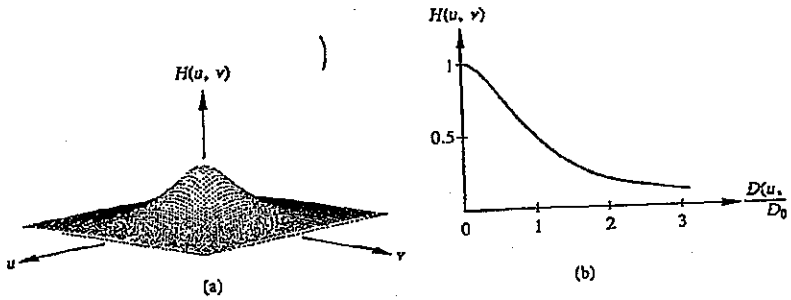


Figure 4.21 (a) A Butterworth lowpass filter. (b) Radial cross sections for $n = 1$.

maximum value of $H(u, v)$. For Eq. (4.3-6), the following simple modification yields the desired value when $D(u, v) = D_0$:

$$H(u, v) = \frac{1}{1 + [\sqrt{2} - 1][D(u, v)/D_0]^{2n}} = \frac{1}{1 + 0.414[D(u, v)/D_0]^{2n}} \quad (4.3\text{-}7)$$

Example: Figure 4.22 shows the results of applying BLPFs (Eq. 4.3-7) to Fig. 4.18(a), with $n = 1$ and D_0 equal to the first five radii shown in Fig. 4.18(b). Note that these images are considerably less blurred than the corresponding results obtained with ideal lowpass filters. The reason is that the "tail" in the BLPF passes a fairly high amount of high-frequency information, thus preserving more of the edge content in the picture. It is also of interest to note that no ringing is evident in any of the images processed with the BLPF, a fact attributed to the filter's smooth transition between low and high frequencies. □

As indicated in Section 4.1, all the filtering results presented in this section were obtained by directly computing the FFT without extending the images to avoid wraparound error. As shown in Fig. 4.22(e), the error is certainly not objectionable since this image is essentially of the same quality as the original, in spite of the fact that convolution with a broad filter was carried out. The reason for this is that the spectrum of $f(x, y)$ falls off rapidly, with 90% of the signal power being contained inside a circle of radius 5. Therefore the amplitude of $F(u, v)$ is relatively small over a large portion of the frequency plane, and these small values attenuate the wraparound error caused by overlaps in the convolution periods. This behavior is typical in practice and often allows us to ignore the error incurred in discrete convolution when the periodic constraints imposed by Eqs. (3.3-33) and (3.3-34) are not satisfied.

Example: The lowpass filtering results given thus far have been with images of good quality in order to illustrate and compare the effect of the filters discussed in this section. Figure 4.23 shows two practical applications of lowpass filtering for image smoothing. The image shown in Fig. 4.23(a) was digitized with only 16 gray levels and, as a consequence, exhibits a considerable amount of false contouring. Figure 4.23(b) is the result of smoothing this image with a lowpass Butterworth filter of order 1. Similarly, Fig. 4.23(d) shows the effect of applying a BLPF to the noisy image of Fig. 4.23(c). It is noted from these examples that lowpass filtering is a cosmetic process that reduces spurious effects at the expense of image sharpness. □

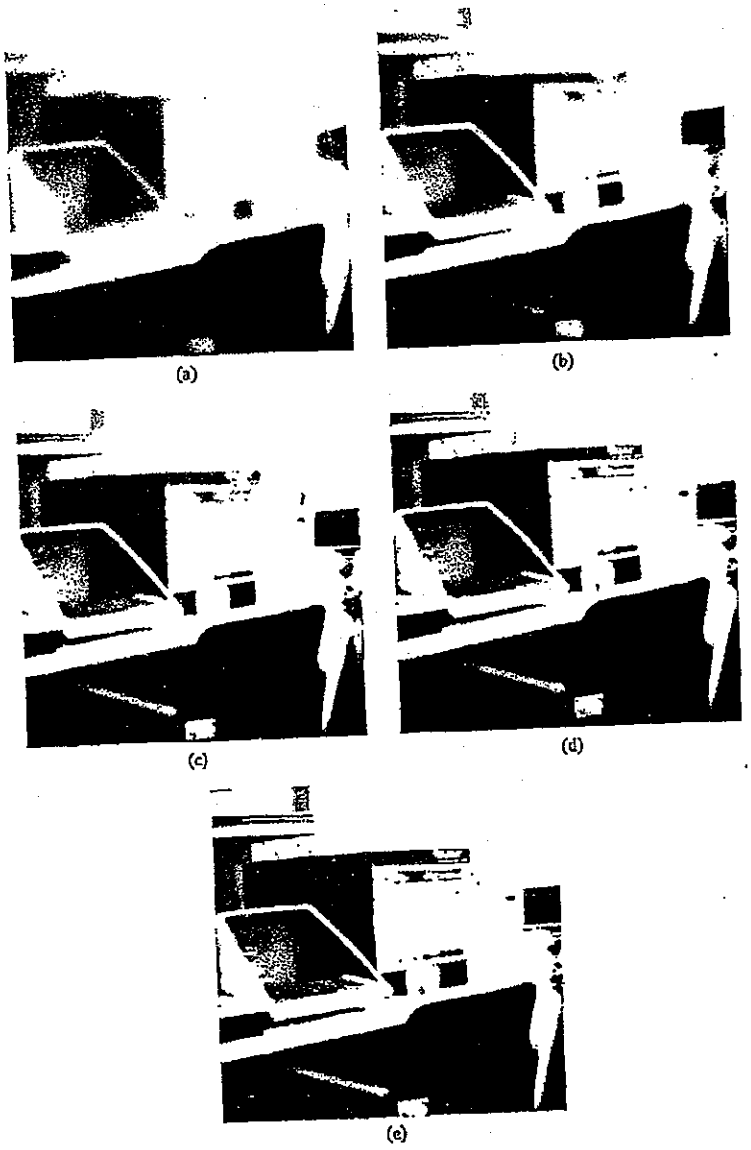


Figure 4.22 Results of applying Butterworth lowpass filters to Fig. 4.18(a). The first five radii shown in Fig. 4.18(b) were used.

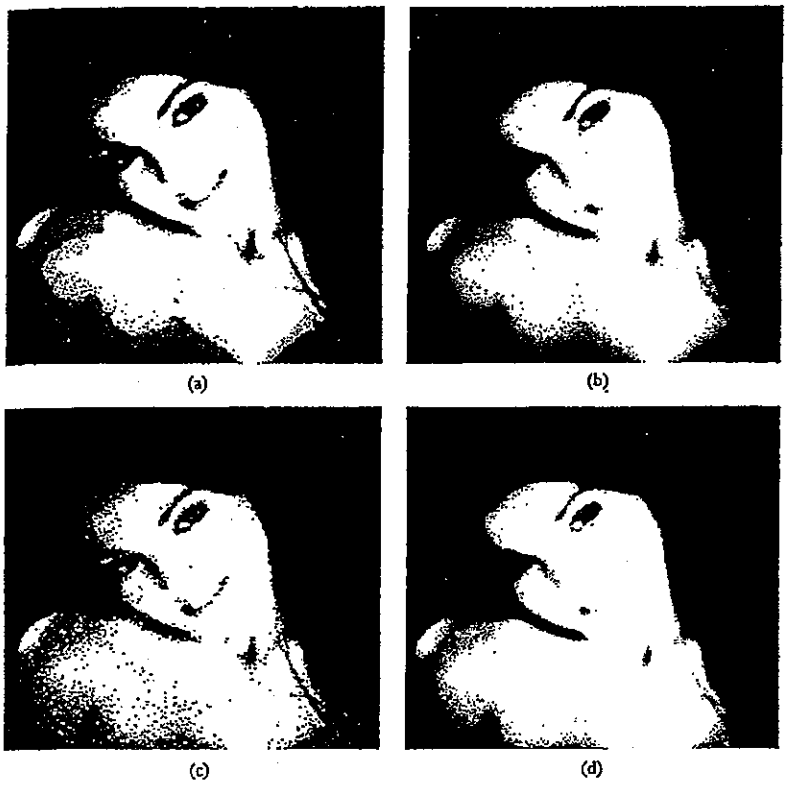


Figure 4.23 Two examples of image smoothing by lowpass filtering (see text).

4.3.4 Averaging of Multiple Images

Consider a noisy image $g(x, y)$ that is formed by the addition of noise $\eta(x, y)$ to an original image $f(x, y)$; that is,

$$g(x, y) = f(x, y) + \eta(x, y), \tag{4.3-8}$$

where it is assumed that at every pair of coordinates (x, y) the noise is uncorrelated and has zero average value. The objective of the following procedure is to obtain a smoothed result by adding a given set of noisy images $\{g_i(x, y)\}$.

If the noise satisfies the constraints just stated, it is a simple problem to show (e.g., see Papoulis [1965]) that if an image $\bar{g}(x, y)$ is formed by averaging M

different noisy images,

$$\bar{g}(x, y) = \frac{1}{M}\sum_{i=1}^{M} g_i(x, y) \tag{4.3-9}$$

then it follows that

$$E\{\bar{g}(x, y)\} = f(x, y) \tag{4.3-10}$$

and

$$\sigma^2_{\bar{g}(x,y)} = \frac{1}{M}\sigma^2_{\eta(x,y)}, \tag{4.3-11}$$

where $E\{\bar{g}(x, y)\}$ is the expected value of $\bar{g}$, $\sigma^2_{\bar{g}(x,y)}$ and $\sigma^2_{\eta(x,y)}$ are the variances of $\bar{g}$ and η, all at coordinates (x, y). The standard deviation at any point in the average image is given by

$$\sigma_{\bar{g}(x,y)} = \frac{1}{\sqrt{M}}\sigma_{\eta(x,y)}. \tag{4.3-12}$$

Equations (4.3-11) and (4.3-12) indicate that as M increases, the variability of the pixel values decreases. Since $E\{\bar{g}(x, y)\} = f(x, y)$, this means that $\bar{g}(x, y)$ will approach $f(x, y)$ as the number of noisy images used in the averaging process increases. In practice, the principal difficulty in using this method lies in being able to register the images so that corresponding pixels line up correctly.

Example: As an illustration of the averaging method, consider the images shown in Fig. 4.24. Part (a) of this figure shows an original image, and part (b) is the same image, with each pixel corrupted by additive Gaussian noise with zero mean and standard deviation equal to 20. Since negative pixel values were not allowed, any $\bar{g}(x, y)$ that was negative as a result of adding noise to $f(x, y)$ was replaced by $|\bar{g}(x, y)|$. In this sense, the noise can only be considered approximately Gaussian. Registration was not a problem in this case because all noisy images were generated from the same source.

Figures 4.24(c) through (h) show the results of using $M = 2$, 5, 10, 25, 50, and 100, respectively, in the averaging process given in Eq. (4.3-9). There is little discernible difference between the original noisy image and the sum of two samples. When $M = 5$, however, the reduction in noise is quite apparent. The same is true of the case where $M = 10$, although some noisy regions are still clearly visible. (Compare, for example, the edges of the robe sleeve and leg nearest the right side of Figs. 4.24a and e.) Averaging 25 copies (Fig. 4.24f) yielded an image that is almost of the same quality and definition as the original. A little noise is still visible in regions such as the sleeve and leg mentioned above, but it is difficult to tell the two images apart by casual observation. The results for $M = 50$ and $M = 100$ are, for all practical purposes, of the same quality as the original image. □

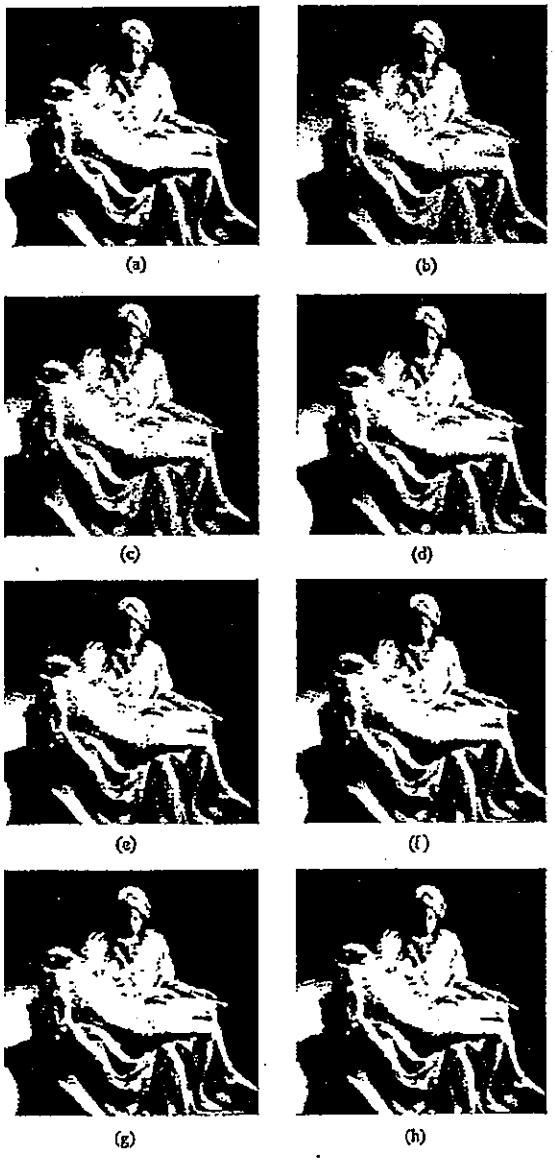


Figure 4.24 Smoothing by superposition of multiple images. (a) Original image. (b) Noisy image. (c) through (h) Results obtained by averaging 2, 5, 10, 25, 50, and 100 copies, respectively.

4.4 IMAGE SHARPENING

Sharpening techniques are useful primarily as enhancement tools for highlighting edges in an image. Following the same format as in Section 4.3, we present below sharpening methods in both the spatial and frequency domains.

4.4.1 Sharpening by Differentiation

It was noted in Section 4.3 that averaging pixels over a region tends to blur detail in an image. Since averaging is analogous to integration, it is natural to expect that differentiation will have the opposite effect and thus sharpen a given image.

The most commonly used method of differentiation in image processing applications is the *gradient*. Given a function $f(x, y)$, the gradient of f at coordinates (x, y) is defined as the *vector*

$$\mathbf{G}[f(x, y)] = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix}. \tag{4.4-1}$$

Two important properties of the gradient are: (1) the vector $\mathbf{G}[f(x, y)]$ points in the direction of the maximum rate of increase of the function $f(x, y)$; and (2) the magnitude of $\mathbf{G}[f(x, y)]$, denoted by $G[f(x, y)]$, and given by

$$G[f(x, y)] = \text{mag}[\mathbf{G}] = [(\partial f/\partial x)^2 + (\partial f/\partial y)^2]^{1/2} \tag{4.4-2}$$

equals the maximum rate of increase of $f(x, y)$ per unit distance in the direction of $\mathbf{G}$.

Equation (4.4-2) is the basis for a number of approaches to image differentiation. It is noted that this expression is in the form of a two-dimensional derivative function and that it is always positive. In practice, the scalar function $G[f(x, y)]$ is commonly referred to as the gradient of f. This terminology will be used throughout the following discussion to avoid having to continually refer to $G[f(x, y)]$ as "the magnitude of the gradient." The reader should, however, keep in mind the basic difference between Eqs. (4.4-1) and (4.4-2).

For a digital image, the derivatives in Eq. (4.4-2) are approximated by differences. One typical approximation is given by the relation

$$G[f(x, y)] \cong \{[f(x, y) - f(x + 1, y)]^2 + [f(x, y) - f(x, y + 1)]^2\}^{1/2} \tag{4.4-3}$$

Similar results are obtained by using absolute values, as follows:

$$G[f(x, y)] \cong |f(x, y) - f(x + 1, y)| + |f(x, y) - f(x, y + 1)|. \tag{4.4-4}$$

This formulation is more desirable for a computer implementation of the gradient. It is also easier to program in assembly language if speed of computation is an essential requirement.

The relationship between pixels in Eqs. (4.4-3) and (4.4-4) is shown in Fig. 4.25(a). For an $N \times N$ image, note that it is not possible to take the gradient for pixels in the last row ($x = N$) or the last column ($y = N$). If an $N \times N$ gradient image is desired, one procedure that can be followed for pixels in these regions is to duplicate the gradients obtained in the previous row when $x = N$ and the previous column when $y = N$.

The above arrangement for approximating the gradient is by no means unique. Another useful approximation, sometimes called the *Roberts gradient*, uses the cross-differences shown in Fig. 4.25(b). This approximation is given by the relation

$$G[f(x, y)] = \{[f(x, y) - f(x + 1, y + 1)]^2 + [f(x + 1, y) - f(x, y + 1)]^2\}^{1/2} \quad (4.4\text{-}5)$$

or, using absolute values,

$$G[f(x, y)] \cong |f(x, y) - f(x + 1, y + 1)| + |f(x + 1, y) - f(x, y + 1)|. \quad (4.4\text{-}6)$$

Note that in all the approximations given above the value of the gradient is proportional to the difference in gray level between adjacent pixels. Thus, as expected, the gradient assumes relatively large values for prominent edges in an image, and small values in regions that are fairly smooth, being zero only in regions that have a constant gray level. These properties of the gradient are illustrated in Fig. 4.26. The digital image shown in Fig. 4.26(a) is composed of two levels. As shown in Fig. 4.26(b), the gradient operation (Eq. 4.4-4 was used) reduces all the constant white regions to zero (black), leaving only the points associated with abrupt changes in gray level (in this case the edge boundaries and the small spot on the upper-right part of the letter T).

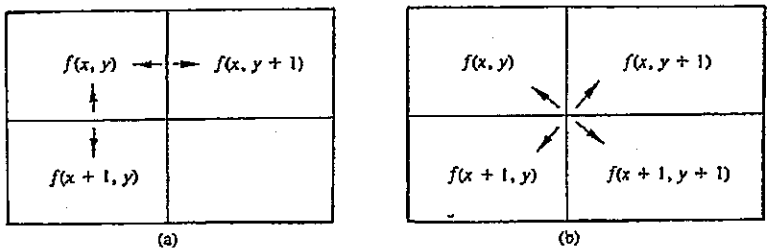
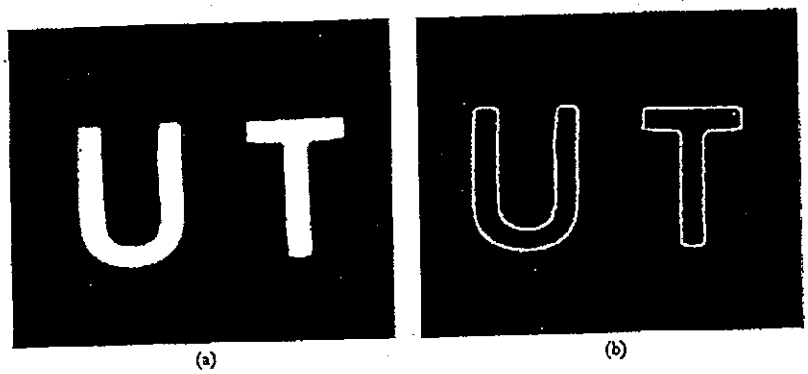


Figure 4.25 Two procedures for computing a two-dimensional, discrete gradient.



(a) (b)

Figure 4.26 (a) A binary image. (b) Result of computing the gradient.

Once a method for approximating the gradient has been selected, there are numerous ways of using the results for generating a gradient image $g(x, y)$. The simplest approach is to let the value of g at coordinates (x, y) be equal to the gradient of f at that point, that is,

$$g(x, y) = G[f(x, y)]. \tag{4.4-7}$$

The principal disadvantage of this method is that all smooth regions in $f(x, y)$ appear dark in $g(x, y)$ because of the relatively small values of the gradient in these regions. One solution to this problem is to form $g(x, y)$ as follows:

$$g(x, y) = \begin{cases} G[f(x, y] & \text{if } G[f(x, y)] \geq T \\ f(x, y) & \text{otherwise,} \end{cases} \tag{4.4-8}$$

where T is a nonnegative threshold. By properly selecting T, it is possible to emphasize significant edges without destroying the characteristics of smooth backgrounds. A variation of this approach where edges are set to a specified gray level L_G is given by

$$g(x, y) = \begin{cases} L_G & \text{if } G[f(x, y)] \geq T \\ f(x, y) & \text{otherwise.} \end{cases} \tag{4.4-9}$$

It is sometimes desirable to study the gray-level variation of edges without interference from the background. This can be accomplished by forming the gradient image as follows:

$$g(x, y) = \begin{cases} G[f(x, y)] & \text{if } G[f(x, y)] \geq T \\ L_B & \text{otherwise,} \end{cases} \tag{4.4-10}$$

where L_B is a specified level for the background.

Finally, if only the location of edges is of interest, the relation

$$g(x, y) = \begin{cases} L_G & \text{if } G[f(x, y)] \geq T \\ L_B & \text{otherwise} \end{cases} \tag{4.4-11}$$

gives a binary gradient picture where the edges and background are displayed in any two specified gray levels.

Example: The types of edge enhancement that can be obtained by using Eq. (4.4-4) and Eqs. (4.4-7) through (4.4-11) are illustrated in Fig. 4.27. Part (a) of the figure shows an original image of moderate complexity. Figure 4.27(b) is the result of using the gradient scheme given by Eq. (4.4-7). Note that a considerable amount of small segments appeared in the resulting image, with the strongest intensities taking place around the border of the aircraft. This is an expected result since the magnitude of the gradient is proportional to changes in gray levels and should be more prominent in regions of an image containing distinct edges.

Figure 4.27(c) was obtained by using Eq. (4.4-8) with $T = 25$, which is approximately 10% of the maximum gray-level value in the original image. The gradient values appear a dark shade of gray because they are displayed on the relatively light background of the image. The important points to note in connection with this figure is that only prominent edges are outlined as a result of using a threshold, and also that the background has not been completely obliterated.

Figure 4.27(d) is the result of using Eq. (4.4-9) with $T = 25$ and $L_G = 255$, the latter being the brightest possible level in the system used to display the results. It is noted that Figs. 4.27(c) and (d) are the same, with the exception that the gradient points exceeding the threshold are much more visible in the latter image.

Figure 4.27(e) was obtained by using Eq. (4.4-10) with the same threshold as above and a background level of $L_B = 0$, which is the darkest possible display level. The principal use of this particular approach is to examine the relative strength of gradient points that exceed the specified threshold. In this case, we see that the outline of the aircraft and the cloud near the bottom of the image are quite prominent in relation to other sections of the picture.

Finally, Fig. 4.27(f) was obtained by using Eq. (4.4-11) with $T = 25$, $L_G = 255$, and $L_B = 0$. This equation is useful for displaying only the gradient points above the specified threshold.

4.4.2 Highpass Filtering

It was shown in Section 4.3.3 that an image can be blurred by attenuating the high-frequency components of its Fourier transform. Since edges and other abrupt changes in gray levels are associated with high-frequency components, image sharpening can be achieved in the frequency domain by a *highpass filtering* process, which attenuates the low-frequency components without disturbing high-frequency information in the Fourier transform.

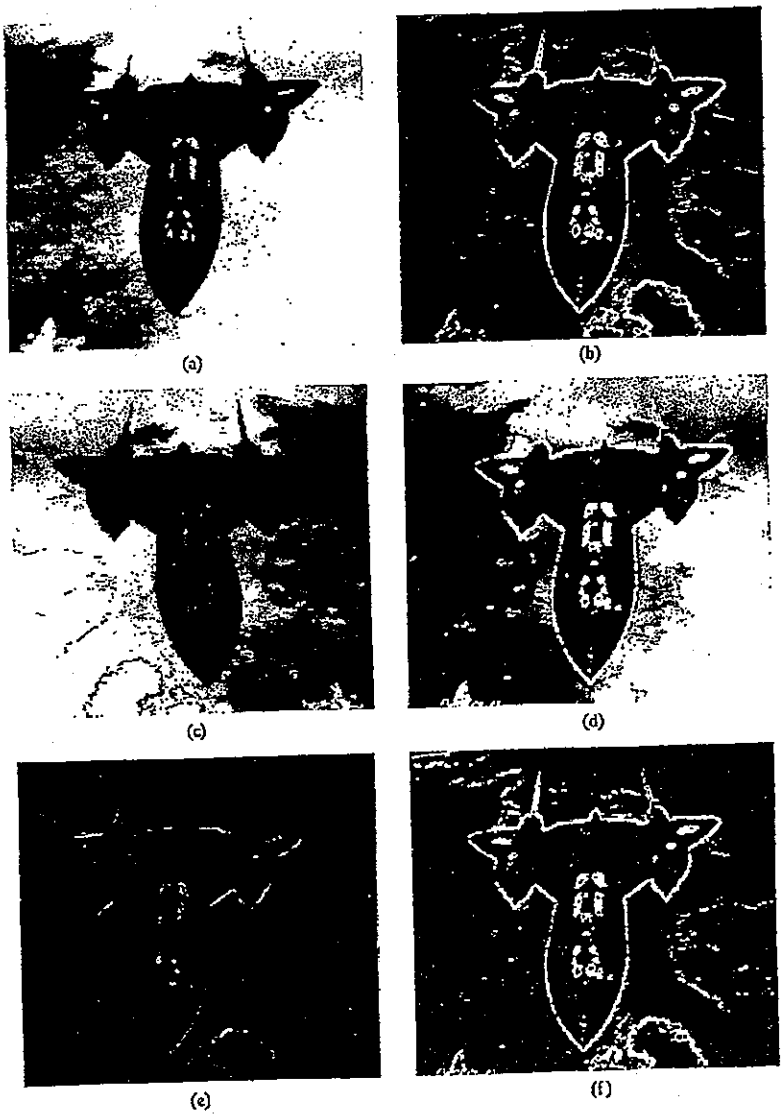


Figure 4.27 Illustration of edge enhancement by gradient techniques.

We consider below the high-frequency counterparts of the filters developed in Section 4.3.3. As before, we will only consider zero-phase-shift filters that are radially symmetric and can be completely specified by a cross section extending as a function of distance from the origin of the Fourier transform.

Ideal filter

A two-dimensional ideal highpass filter (IHPF) is one whose transfer function satisfies the relation

$$H(u, v) = \begin{cases} 0 & \text{if } D(u, v) \leq D_0 \\ 1 & \text{if } D(u, v) > D_0, \end{cases} \tag{4.4-12}$$

where D_0 is the cut-off distance measured from the origin of the frequency plane, and $D(u, v)$ is given by Eq. (4.3-5). A perspective plot and cross section of the IHPF function are shown in Fig. 4.28. It is noted that this filter is just the opposite of the ideal lowpass filter discussed in Section 4.4.3 since it completely attenuates all frequencies inside a circle of radius D_0 while passing, without attenuation, all frequencies outside the circle. As in the case of the ideal lowpass filter, the IHPF is not physically realizable.

Butterworth filter

The transfer function of the Butterworth highpass filter (BHPF) of order n and with cut-off frequency locus at a distance D_0 from the origin is defined by the relation

$$H(u, v) = \frac{1}{1 + [D_0 / D(u, v)]^{2n}}, \tag{4.4-13}$$

where $D(u, v)$ is given by Eq. (4.3-5). A perspective plot and cross section of the BHPF function are shown in Fig. 4.29.

Note that when $D(u, v) = D_0$, $H(u, v)$ is down to ½ of its maximum value. As in the case of the Butterworth lowpass filter, it is common practice to select

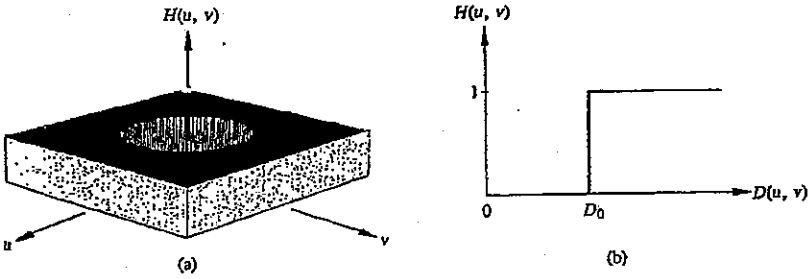


Figure 4.28 Perspective plot and radial cross section of ideal highpass filter.

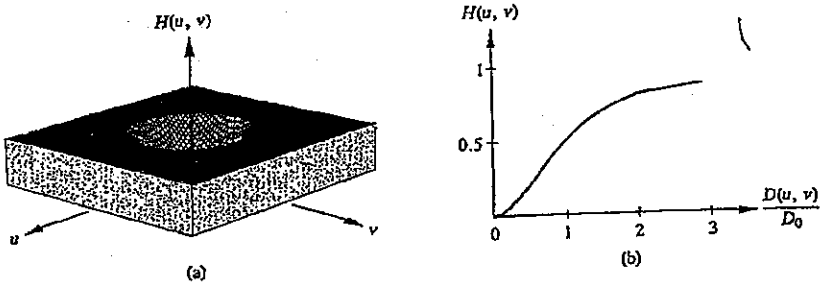


Figure 4.29 Perspective plot and radial cross section of Butterworth highpass filter for $n = 1$.

the cut-off frequency locus at points for which $H(u, v)$ is down to $1/\sqrt{2}$ of its maximum value. Equation (4.4-13) is easily modified to satisfy this constraint by using the following scaling:

$$H(u, v) = \frac{1}{1 + [\sqrt{2} - 1][D_0/D(u, v)]^{2n}} = \frac{1}{1 + 0.414[D_0/D(u, v)]^{2n}} \tag{4.4-14}$$

Example: Figure 4.30(a) shows a chest x-ray that was poorly developed, and Fig. 4.30(b) shows the image after it was processed with a highpass Butterworth filter of order 1. Only the edges are predominant in this image because the low-frequency components were severely attenuated, thus making different (but smooth) gray-level regions appear essentially the same.

A technique often used to alleviate this problem consists of adding a constant to a highpass filter transfer function in order to preserve the low-frequency components. This, of course, amplifies the high-frequency components to values that are higher than in the original transform. This technique, called *high-frequency emphasis*, is illustrated in Fig. 4.30(c). Note that the image in this case has a little better tonality, particularly in the lower-left part of the photograph.

Although high-frequency emphasis preserves the low-frequency components the proportionally larger high-frequency terms tend to obscure the result, as shown by the small gain in quality from Fig. 4.30(b) to Fig. 4.30(c). A technique often used to compensate for this problem is to do some post-filtering processing to redistribute the gray levels. Histogram equalization is ideally suited for this because of its simplicity. Figure 4.30(d) shows the significant improvement that can be obtained by histogram-equalizing an image that has been processed by high-frequency emphasis.

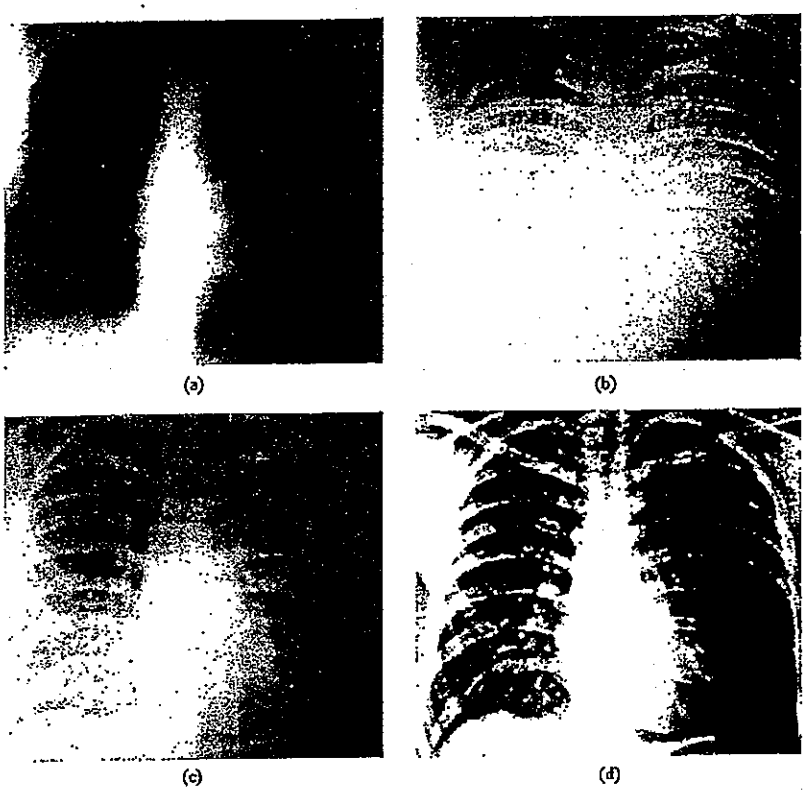


Figure 4.30 Example of highpass filtering. (a) Original image. (b) Image processed with a highpass Butterworth filter. (c) Result of high-frequency emphasis. (d) High-frequency emphasis and histogram equalization. (From Hall *et al.* [1971].)

4.5 ENHANCEMENT BASED ON AN IMAGE MODEL

The illumination–reflectance model introduced in Section 2.2 can be used as the basis for a frequency-domain procedure that is useful for improving the appearance of an image by simultaneous brightness range compression and contrast enhancement. From the discussion in Section 2.2 we have that an image $f(x, y)$ can be expressed in terms of its illumination and reflectance components by means of the relation

$$f(x, y) = i(x, y)r(x, y). \tag{4.5-1}$$

Equation (4.5-1) cannot be used directly in order to operate separately on the frequency components of illumination and reflectance because the Fourier transform of the product of two functions is not separable; in other words,

$$\mathfrak{F}\{f(x, y)\} \neq \mathfrak{F}\{i(x, y)\}\mathfrak{F}\{r(x, y)\}$$

Suppose, however, that we let

$$\begin{aligned} z(x, y) &= \ln f(x, y) \\ &= \ln i(x, y) + \ln r(x, y). \end{aligned} \tag{4.5-2}$$

Then, it follows that

$$\begin{aligned} \mathfrak{F}\{z(x, y)\} &= \mathfrak{F}\{\ln f(x, y)\} \\ &= \mathfrak{F}\{\ln i(x, y)\} + \mathfrak{F}\{\ln r(x, y)\} \end{aligned} \tag{4.5-3}$$

or

$$Z(u, v) = I(u, v) + R(u, v), \tag{4.5-4}$$

where $I(u, v)$ and $R(u, v)$ are the Fourier transforms of $\ln i(x, y)$ and $\ln r(x, y)$, respectively.

If we process $Z(u, v)$ by means of a filter function $H(u, v)$, it follows from Eq. (4.1-2) that

$$\begin{aligned} S(u, v) &= H(u, v)Z(u, v) \\ &= H(u, v)I(u, v) + H(u, v)R(u, v), \end{aligned} \tag{4.5-5}$$

where $S(u, v)$ is the Fourier transform of the result. In the spatial domain, we have the relation

$$\begin{aligned} s(x, y) &= \mathfrak{F}^{-1}\{S(u, v)\} \\ &= \mathfrak{F}^{-1}\{H(u, v)I(u, v)\} + \mathfrak{F}^{-1}\{H(u, v)R(u, v)\}. \end{aligned} \tag{4.5-6}$$

By letting

$$i'(x, y) = \mathfrak{F}^{-1}\{H(u, v)I(u, v)\} \tag{4.5-7}$$

and

$$r'(x, y) = \mathfrak{F}^{-1}\{H(u, v)R(u, v)\}, \tag{4.5-8}$$

we can express Eq. (4.5-6) in the form

$$s(x, y) = i'(x, y) + r'(x, y). \tag{4.5-9}$$

Finally, since $z(x, y)$ was formed by taking the logarithm of the original imag

$f(x, y)$, we now perform the inverse operation to obtain the desired enhanced image $g(x, y)$; that is,

$$\begin{aligned} g(x, y) &= \exp\{s(x, y)\} \\ &= \exp\{i'(x, y)\} \cdot \exp\{r'(x, y)\} \\ &= i_0(x, y) r_0(x, y), \end{aligned} \tag{4.5-10}$$

where

$$i_0(x, y) = \exp\{i'(x, y)\} \tag{4.5-11}$$

and

$$r_0(x, y) = \exp\{r'(x, y)\} \tag{4.5-12}$$

are the illumination and reflectance components of the output image.

The enhancement approach using the foregoing concepts is summarized in Fig. 4.31. This method is based on a special case of a class of systems known as *homomorphic systems*. In this particular application, the key to the approach is the fact that separation of the illumination and reflectance components is achieved in the form shown in Eq. (4.5-4). It is then possible for the *homomorphic filter function* $H(u, v)$ to operate on these components separately, as indicated in Eq. (4.5-5).

The illumination component of an image is generally characterized by slow spatial variations. The reflectance component, on the other hand, tends to vary abruptly, particularly at the junctions of very dissimilar objects. These characteristics lead us to associate the low frequencies of the Fourier transform of the logarithm of an image with illumination, and the high frequencies with reflectance. Although this is a rough approximation, it can be used to advantage in image enhancement.

Illumination is directly responsible for the dynamic range achieved by the pixels in an image. Similarly, contrast is a function of the reflective nature of the objects in the image. A good deal of control can be gained over these components by using a homomorphic filter. This requires specification of a filter function $H(u, v)$ that will affect the low- and high-frequency components of the Fourier transform in different ways. A cross section of such a function is shown in Fig. 4.32. A complete specification of $H(u, v)$ is obtained by rotating the cross section 360° about the vertical axis. If the parameters γ_L and γ_H are chosen so that $\gamma_L < 1$ and $\gamma_H > 1$, the filter function shown in Fig. 4.32 will tend to decrease the low frequencies and amplify the high frequencies. The net result is simultaneous dynamic range compression and contrast enhancement.

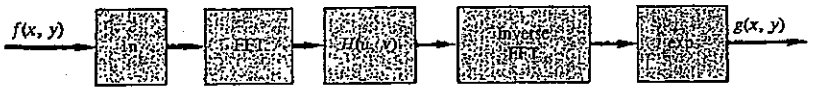


Figure 4.31 Homomorphic filtering approach for image enhancement.

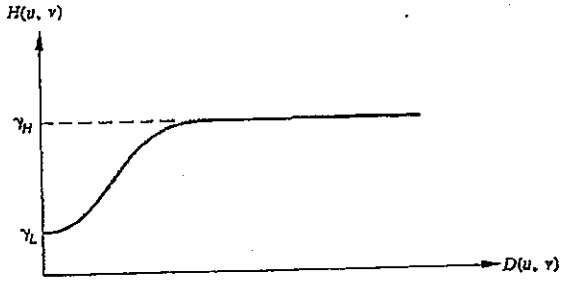


Figure 4.32 Cross section of a circularly symmetric filter function for use in homomorphic filtering. $D(u, v)$ is the distance from the origin.

Example: Figure 4.33 is typical of the results that can be obtained with the homomorphic filter function shown in Fig. 4.32. We note in the original image, Fig. 4.33(a), that the details inside the room are obscured by the glare from the outside walls. Figure 4.33(b) shows the result of processing this image by homomorphic filtering with $\gamma_L = 0.5$ and $\gamma_H = 2.0$ in the above filter function. A reduction of dynamic range in the brightness, together with an increase in contrast, brought out the details of objects inside the room and balanced the levels of the outside wall. □

4.6 GENERATION OF SPATIAL MASKS FROM FREQUENCY-DOMAIN SPECIFICATIONS

As indicated in Section 4.1, speed and simplicity of implementation are important features of spatial masks for image processing. On the other hand, certain filtering functions (such as lowpass filtering) are more conveniently specified in the frequency domain. In this section we develop a method for generating spatial masks that approximate (in a least-square-error sense) a given frequency-domain filter.

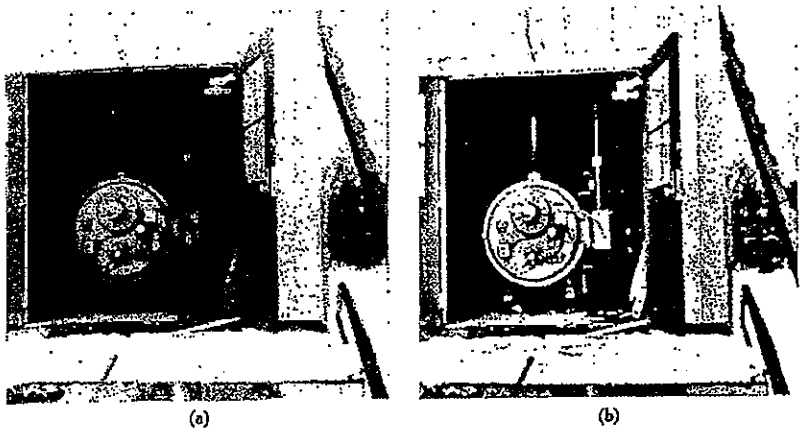
As discussed in Section 4.1, the filtering process in the frequency domain is based on the equation

$$G(u, v) = H(u, v)F(u, v), \tag{4.6-1}$$

where $F(u, v)$ and $G(u, v)$ are the Fourier transforms of the input and output images, respectively, and $H(u, v)$ is the filter transfer function. From the convolution theorem (Section 3.3.8), we know that Eq. (4.6-1) can be implemented in the spatial domain by the expression

$$g(x, y) = \sum_{i=0}^{N-1} \sum_{k=0}^{N-1} h(x - i, y - k)f(i, k) \tag{4.6-2}$$

with $x = 0, 1, 2, \ldots, N - 1$ and $y = 0, 1, 2, \ldots, N - 1$. For simplicity in notation, it is assumed that we are working with square image arrays. Also, it is



(a) (b)

Figure 4.33 (a) Original image. (b) Image processed by homomorphic filtering to achieve simultaneous dynamic range compression and contrast enhancement. (From Stockham [1972].)

understood that all functions have been properly extended, as discussed in Section 3.3.8, under Convolution.

In Eq. (4.6-2), h is the spatial representation of the filter (i.e., the inverse Fourier transform of $H(u, v)$), f is the input image, and g is the filtered image. We often refer to h as a *spatial convolution mask*. If this mask is of size $N \times N$, the result given in Eq. (4.6-2) is identical to taking the inverse Fourier transform of $G(u, v)$ in Eq. (4.6-1).

Since H is the Fourier transform of h, it follows from Eq. (3.2-9) that

$$H(u, v) = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} h(x, y) \exp[-j2\pi(ux + vy)/N] \tag{4.6-3}$$

for $u, v = 0, 1, 2, \ldots, N - 1$. Suppose, however, that we restrict $h(x, y)$ to be zero for values of $x > n$ and $y > n$, with $n < N$. This in effect creates an $n \times n$ convolution mask $\hat{h}$ with Fourier transform

$$\hat{H}(u, v) = \frac{1}{N} \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} \hat{h}(x, y) \exp[-j2\pi(ux + vy)/N] \tag{4.6-4}$$

for $u, v = 0, 1, 2, \ldots, N - 1$. The objective in the following discussion is to find the coefficients of $\hat{h}(x, y)$ so that the error quantity

$$e^2 = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} |\hat{H}(u, v) - H(u, v)|^2 \tag{4.6-5}$$

is minimized, where $|\cdot|$ designates the complex magnitude.

Equation (4.6-4) can be expressed in the following matrix form:

$$\hat{\mathbf{H}} = \mathbf{C}\hat{\mathbf{h}}, \tag{4.6-6}$$

where $\hat{\mathbf{H}}$ is a column vector of order N^2 containing the terms for $\hat{H}(u, v)$ in some order, $\hat{\mathbf{h}}$ is a column vector of order n^2 containing the elements of $\hat{h}(x, y)$ in some order, and $\mathbf{C}$ is an $N^2 \times n^2$ matrix of exponential terms whose positions are determined by the ordering in $\hat{\mathbf{H}}$ and $\hat{\mathbf{h}}$. A simple procedure for generating the elements $\hat{\mathbf{H}}(i)$, $i = 0, 1, 2, \ldots, N^2 - 1$, of the vector $\hat{\mathbf{H}}$ from $\hat{H}(u, v)$ is as follows:

$$\hat{H}(u, v) \Rightarrow \hat{\mathbf{H}}(i) \tag{4.6-7}$$

for $i = uN + v$, with $u, v = 0, 1, 2, \ldots, N - 1$. It is noted that stepping through the rows of $\hat{H}(u, v)$ by letting $u = 0$, $v = 0, 1, 2, \ldots, N - 1$; $u = 1$, $v = 0, 1, 2, \ldots, N - 1$; and so on, corresponds to forming the first N elements of $\hat{\mathbf{H}}$ from the first row of $\hat{H}(u, v)$, the next N elements from the second row, and so forth. The elements of $\hat{\mathbf{h}}$, denoted by $\hat{h}(k)$, $k = 0, 1, 2, \ldots, n^2 - 1$, are similarly formed by letting

$$\hat{h}(x, y) \Rightarrow \hat{h}(k) \tag{4.6-8}$$

for $k = xn + y$, with $x, y = 0, 1, 2, \ldots, n - 1$. Finally, the corresponding elements of the matrix $\mathbf{C}$, denoted by $C(i, k)$, are generated from the exponential terms

$$\frac{1}{N}\exp[-j2\pi(ux + vy)/N] \Rightarrow C(i, k) \tag{4.6-9}$$

for $i = uN + v$ and $k = xn + y$, with $u, v = 0, 1, 2, \ldots, N - 1$, and $x, y = 0, 1, 2, \ldots, N - 1$.

Using matrix notation, Eq. (4.6-5) can be written in the form

$$\begin{aligned} e^2 &= (\hat{\mathbf{H}} - \mathbf{H})^*(\hat{\mathbf{H}} - \mathbf{H}) \\ &= \|\hat{\mathbf{H}} - \mathbf{H}\|^2 \\ &= \|\mathbf{C}\hat{\mathbf{h}} - \mathbf{H}\|^2, \end{aligned} \tag{4.6-10}$$

where $*$ is the conjugate transpose, $\|\cdot\|$ is the complex Euclidean norm, and $\mathbf{H}$ is a vector formed from $H(u, v)$ in the manner explained above. We find the minimum of e^2 with respect to $\hat{\mathbf{h}}$ by taking the partial derivative and equating it to the zero vector:

$$\frac{\partial e^2}{\partial \hat{\mathbf{h}}} = 2\mathbf{C}^*(\mathbf{C}\hat{\mathbf{h}} - \mathbf{H}) = 0 \tag{4.6-11}$$

or

$$\begin{aligned} \hat{\mathbf{h}} &= (\mathbf{C}^*\mathbf{C})^{-1}\mathbf{C}^*\mathbf{H} \\ &= \mathbf{C}^{\#}\mathbf{H} \end{aligned} \tag{4.6-12}$$

where the matrix $\mathbf{C}^{\#} = (\mathbf{C}^*\mathbf{C})^{-1}\mathbf{C}^*$ is often called the *Moore–Penrose generalized inverse* (Noble [1969]).

Equation (4.6-12) yields the necessary minimum-error coefficients to construct an $n \times n$ convolution mask $\hat{h}(x, y)$ from a specified $N \times N$ filter function $H(u, v)$ in the frequency domain. In general, the elements of $\hat{h}(x, y)$ are complex quantities. However, it is easily shown that if the frequency-domain filter function is real and symmetric (like all the filters discussed in this chapter), then $\hat{h}(x, y)$ will also be real and symmetric.

Example: As an illustration of the method just developed, the test pattern shown in Fig. 4.34(a) was filtered using a Butterworth lowpass filter to produce the blurred image shown in Fig. 4.34(b). A 9×9 convolution mask was generated using Eq.

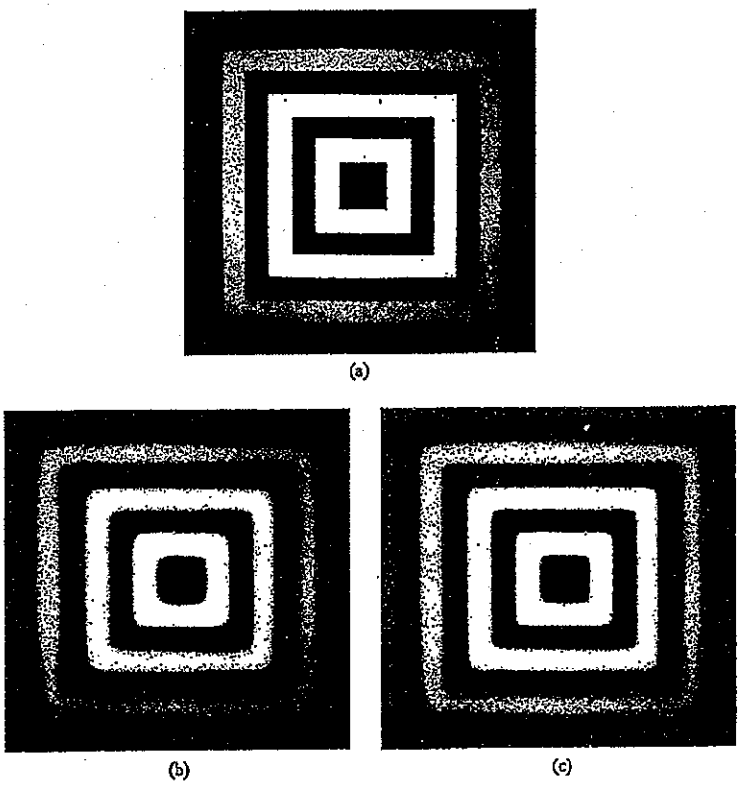


Figure 4.34 (a) Original image. (b) Blurred image obtained with a Butterworth lowpass filter of order 1 in the frequency domain. (c) Image blurred spatially by a 9×9 convolution mask obtained using Eq. (4.6-12). (From Meyer and Gonzalez [1983].)

(4.6-12) and applied to the original image. The result, shown in Fig. 4.34(c), is slightly less blurred than that obtained by using the complete filter in the frequency domain. This is expected since the spatial-domain process with $n < N$ is only an approximation in the least-square-error sense. □

4.7 PSEUDO-COLOR IMAGE PROCESSING

Attention has been focused thus far on processing techniques for monochrome images. A relatively recent and potentially powerful area of digital image processing is the use of pseudo-color for image display and enhancement. The motivation for using color in image processing is provided by the fact that the human eye can discern thousands of color shades and intensities. This is in sharp contrast with the eye's relatively poor performance with gray levels where, as indicated in Section 2.1, only one to two dozen shades of gray are detectable at any one point in an image by the average observer. The reader needs only to turn off the color next time he views a TV set in order to verify the superior performance of the eye when interpreting color versus monochrome information.

The basic idea behind the use of pseudo-color should be carefully distinguished from techniques known as false-color processing. The latter are analogous to true-color photography, with the exception that they may deal with light bands that are outside the visible spectrum. Infrared photography is an example of this, where interest lies not in true color fidelity, but rather on information that is most evident in the infrared spectrum. In pseudo-color, the situation is fundamentally different in that processing starts out with a monochrome image. The objective is then to assign a color to each pixel based, for example, on its intensity. The range of techniques for color assignment are limited only by the capabilities of the display system and the ingenuity of the user. As will be seen in the following discussion, even some straightforward techniques for color coding can sometimes bring out information that is often difficult to detect and interpret in a monochrome image.

4.7.1 Color Fundamentals

Although the process followed by the human brain in perceiving color is a physiopsychological phenomenon that is not yet fully understood, the physical nature of color can be expressed on a formal basis supported by experimental and theoretical results.

In 1666, Sir Isaac Newton discovered that when a beam of sunlight is passed through a glass prism the emerging beam of light is not white, but consists instead of a continuous spectrum of colors ranging from violet at one end to red at the other. As shown in Plate I, the color spectrum may be divided into six broad regions: violet, blue, green, yellow, orange, and red. When viewed in full color (Plate II) no color in the spectrum ends abruptly, but rather we have a situation where each color blends smoothly into the next.

Basically, the colors we perceive in an object are determined by the nature of the light reflected from the object. As illustrated in Plate II, visible light is composed

of a relatively narrow band of frequencies in the electromagnetic energy spectrum. A body that reflects light that is relatively balanced in all visible wavelengths appears white to the observer. On the other hand, a body that favors reflectance in a limited range of the visible spectrum will exhibit some shades of color. For example, green objects reflect light with wavelengths primarily in the 500 to 570 nm (10^{-9} m) range, while absorbing most of the energy at other wavelengths.

Due to the structure of the human eye, all colors are seen as variable combinations of the three so-called *primary colors* red (R), green (G), and blue (B). For the purpose of standardization, the CIE (Commission Internationale de l'Eclairage—the International Commission on Illumination) designated in 1931 the following specific wavelength values to the three primary colors: blue = 435.8 nm, green = 546.1 nm, and red = 700 nm. We note from Plate II, however, that no single color may be called red, green, or blue. Thus once we have settled on three specific color wavelengths for the purpose of standardization, one needs to realize that these three fixed RGB components acting alone cannot generate all spectrum colors. This is important because use of the word "primary" has been widely misinterpreted to mean that the three standard primaries, when mixed in various intensity proportions can produce all visible colors. This is not true, unless the wavelength is also allowed to vary.

The primary colors can be added to produce the *secondary* colors of light—magenta (red plus blue), cyan (green plus blue), and yellow (red plus green). Mixing the three primaries, or a secondary with its opposite primary color, in the right intensities produces white light. This is shown in Plate III(a), which also illustrates the three primary colors and their combinations to produce the secondary colors.

It is important to differentiate between the primary colors of light and the primary colors of pigments or colorants. In the latter, a primary color is defined as one that subtracts or absorbs a primary color of light and reflects or transmits the other two. Therefore the primary colors of pigments are magenta, cyan, and yellow, while the secondary colors are red, green, and blue. These colors are shown in Plate III(b). It is noted that a proper combination of the three pigment primaries, or a secondary with its opposite primary, produces black.

Color television reception is an example of the additive nature of light colors. The interior of many color TV tubes is composed of a large array of triangular dot patterns of electron-sensitive phosphor. When excited, each dot in a triad is capable of producing light in one of the primary colors. The intensity of the red-emitting phosphor dots is modulated by an electron gun inside the tube, which generates pulses corresponding to the "red energy" seen by the TV camera. The green and blue phosphor dots in each triad are modulated in the same manner. The effect, viewed on the television receiver, is that the three primary colors from each phosphor triad are "added" together and received by the color-sensitive cones in the eye, and a full-color image is perceived. Thirty successive image changes per second in all three colors complete the illusion of a continuous image display on the screen.

The characteristics generally used to distinguish one color from another are *brightness; hue,* and *saturation.* Brightness refers to intensity. Hue is an attribute

associated with the dominant wavelength in a mixture of light waves. Thus hue represents dominant color as perceived by an observer; when we call an object red, orange, or yellow we are specifying its hue. Saturation refers to relative purity or the amount of white light mixed with a hue. The pure spectrum colors are fully saturated. Colors such as pink (red and white) and lavender (violet and white) are less saturated, with the degree of saturation being inversely proportional to the amount of white light added.

Hue and saturation taken together are called *chromaticity*, and therefore a color may be characterized by its brightness and chromaticity. The amounts of red, green, and blue needed to form any given color are called the *tristimulus* values and are denoted, respectively, by X, Y, and Z. A color is then specified by its *trichromatic coefficients*, defined as

$$x = \frac{X}{X + Y + Z} \tag{4.7-1}$$

$$y = \frac{Y}{X + Y + Z} \tag{4.7-2}$$

and

$$z = \frac{Z}{X + Y + Z}. \tag{4.7-3}$$

It is evident from these equations that

$$x + y + z = 1. \tag{4.7-4}$$

Given any wavelength of light in the visible spectrum, the tristimulus values needed to produce the color corresponding to that wavelength can be obtained directly from curves or tables that have been compiled from extensive experimental results (Walsh [1958], Kiver [1965]).

Another approach for specifying colors is the *chromaticity diagram* (Plate IV), which shows color composition as a function of x (red) and y (green). For any value of x and y, the corresponding value of z (blue) is obtained from Eq. (4.7-4) by noting that $z = 1 - (x + y)$. The point marked "Green" in Plate IV, for example, has approximately 62% green and 25% red content. It then follows from Eq. (4.7-4) that the composition of blue is approximately 13%.

The positions of the various spectrum colors, from violet at 380 nm to red at 780 nm, are indicated around the boundary of the tongue-shaped chromaticity diagram. These are the pure colors shown in the spectrum of Plate II. Any point not actually on the boundary but within the diagram represents some mixture of spectrum colors. The point of equal energy shown in Plate IV corresponds to equal fractions of the three primary colors; it represents the CIE standard for white light. Any point located

on the boundary of the chromaticity chart is said to be completely saturated. As a point leaves the boundary and approaches the point of equal energy, more white light is added to the color and it becomes less saturated. The saturation at the point of equal energy is zero.

The chromaticity diagram is useful for color mixing because a straight line segment joining any two points in the diagram defines all the different color variations that can be obtained by combining these two colors additively. Consider, for example, a straight line drawn from the Red to the Green points shown in Plate IV. If there is more red light than green light, the exact point representing the new color will be on the line segment, but it will be closer to the red point than to the green point. Similarly, a line drawn from the point of equal energy to any point on the boundary of the chart will define all the shades of particular spectrum color.

Extension of the above procedure to three colors is straightforward. To determine the range of colors that can be obtained from any three given colors in the chromaticity diagram, we simply draw connecting lines to each of the three color points. The result is a triangle, and any color inside the triangle can be produced by various combinations of the three initial colors. It is noted that a triangle with vertices at any three *fixed* colors does not enclose the entire color region in Plate IV. This supports graphically the remark made earlier that not all colors can be obtained with three single primaries.

In the following discussion, the intensity of the three primary colors at coordinates (x, y) of an image will be denoted by $I_R(x, y)$, $I_G(x, y)$, and $I_B(x, y)$. Thus each pixel in a color image will be considered as an additive combination of these three intensity values at the coordinates of the pixel.

4.7.2 Density Slicing

The technique of *density* (or *intensity*) *slicing* and color coding is one of the simplest examples of pseudo-color image processing. If an image is viewed as a two-dimensional intensity function (see Section 1.2), the method can be interpreted as one of placing planes parallel to the coordinate plane of the image; each plane then "slices" the function in the area of intersection. Figure 4.35 shows a simple example of this where a plane at $f(x, y) = l_i$ is used to slice a function into two levels. The term "density slicing" arises from calling the gray levels densities, a terminology that is commonly associated with this particular method.

It is evident that if a different color is assigned to each side of the plane shown in Fig. 4.35, then any pixel whose gray level is above the plane will be coded with one color, while any pixel below the plane will be coded with the other. Levels that lie on the plane itself may be arbitrarily assigned one of the two colors. The result of this scheme would produce a two-color image whose relative appearance can be controlled by moving the slicing plane up and down the gray-level axis.

In general, the technique may be summarized as follows. Suppose that M planes are defined at levels $l_1, l_2, \ldots, l_M$, and let l_0 represent black $[f(x, y) = 0]$ and l_L represent white $[f(x, y) = L]$. Then, assuming that $0 < M < L$, the M planes

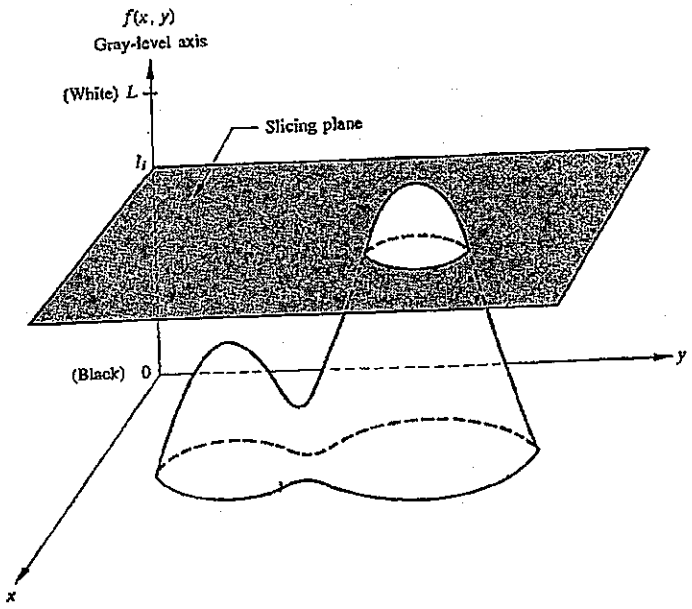


Figure 4.35 Geometrical interpretation of the density-slicing technique.

partition the gray scale into $M + 1$ regions and color assignments are made according to the relation:

$$f(x, y) = c_k \qquad \text{if } f(x, y) \in R_k, \tag{4.7-5}$$

where c_k is the color associated with the kth region R_k defined by the partitioning planes.

It is important to note that the idea of planes is useful primarily for a geometrical interpretation of the density-slicing technique. An alternative representation is shown in Fig. 4.36, which defines the same mapping as Fig. 4.35. According to the mapping function shown in Fig. 4.36, any input gray level is assigned one of two colors, depending on whether it is above or below the value of l_i. When more levels are used, the mapping function assumes a staircase form. This type of mapping is a special case of the approach discussed in the following section.

Example: An example of density slicing is shown in Plate V. Part (a) is a monochrome image of the Picker Thyroid Phantom (a radiation test pattern), and Plate V(b) is the result of density slicing this image into eight color regions. It is noted that

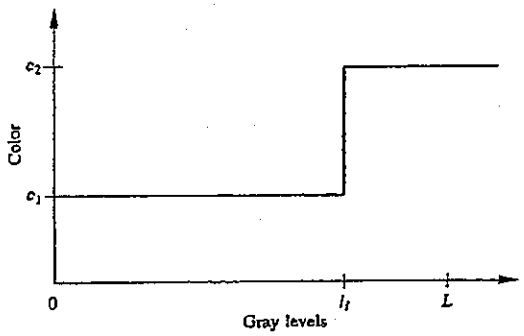


Figure 4.36 An alternate representation of the density-slicing method.

regions that appear of constant intensity in the monochrome image are really quite variable, as shown by the various colors in the sliced image. The left lobe, for instance, is a dull gray in the monochrome image, and it is difficult to pick out variations in intensity. By contrast, the color image clearly shows eight different regions of constant intensity, one for each of the colors used. □

4.7.3 Gray-Level-to-Color Transformations

It is possible to specify other types of transformations that are more general and thus are capable of achieving a wider range of pseudo-color enhancement results than the simple density-slicing technique discussed in the previous section. An approach that is particularly attractive is shown in Fig. 4.37. Basically, the idea underlying this approach is to perform three independent transformations on the gray levels of any input pixel. The three results are then fed separately into the red, green, and blue guns of a color television monitor. This produces a composite image whose color content is modulated by the nature of the transformation function. It should be kept in mind that these are transformations on the gray-level values of an image and that they are not functions of position.

As indicated in the previous section, the method shown in Fig. 4.36 is a special case of the technique just described. The generation of colors in Fig. 4.36 is accomplished by piecewise linear functions of the gray levels. On the other hand, the method discussed in this section can be based on smooth, nonlinear functions that, as might be expected, gives the technique considerably more flexibility. This is demonstrated by the following example.

Example: Plate VI(a) shows a composite monochrome image consisting of two images of luggage obtained from an airport x-ray scanning system. The image on

this allowed us to "see" through the explosives. The background mappings were about the same as those used for Plate VI(b), producing almost identical color assignments. □

4.7.4 A Filtering Approach

Figure 4.39 shows a color-coding scheme that is based on frequency-domain operations. The idea depicted in this figure is the same as the basic filtering approach discussed earlier in this chapter, with the exception that the Fourier transform of an image is modified independently by three filter functions to produce three images that can be fed into the red, green, and blue inputs of a color monitor. Consider, for example, the sequence of steps followed in obtaining the image for the red channel. The Fourier transform of the input image is altered by using a specified filter function. The processed image is then obtained by using the inverse Fourier transform. This can then be followed by additional processing (such as histogram equalization) before the image is fed into the red input of the monitor. Similar comments apply to the other two paths in Fig. 4.39.

The objective of this color-processing technique is to color-code regions of an image based on frequency content! A typical filtering approach is to use lowpass, bandpass (or bandreject), and highpass filters to obtain three ranges of frequency components. We have already discussed lowpass and highpass filters. Bandreject and bandpass filters are an extension of these concepts.

A simple approach for generating filters that reject or attenuate frequencies about a circular neighborhood of a point (u_0, v_0) is to perform a translation of coordinates for the highpass filters discussed in Section 4.4.2. We illustrate the procedure for the ideal filter.

An ideal bandreject filter (IBRF), which suppresses all frequencies in a neighborhood of radius D_0 about a point (u_0, v_0), is given by the relation

$$H(u, v) = \begin{cases} 0 & \text{if } D(u, v) \leq D_0 \\ 1 & \text{if } D(u,v) > D_0, \end{cases} \tag{4.7-6}$$

where

$$D(u, v) = [(u - u_0)^2 + (v - v_0)^2]^{1/2} \tag{4.7-7}$$

Note that Eq. 4.7-6 is identical in form to Eq. (4.4-12), but the distance function $D(u, v)$ is computed about the point (u_0, v_0) instead of the origin.

Due to the symmetry of the Fourier transform, band rejection that is not about the origin must be carried out in symmetric *pairs* in order to obtain meaningful results. In the case of the ideal filter we modify Eq. (4.7-6) as follows:

$$H(u, v) = \begin{cases} 0 & \text{if } D_1(u, v) \leq D_0 \text{ or } D_2(u, v) \leq D_0 \\ 1 & \text{otherwise,} \end{cases} \tag{4.7-8}$$

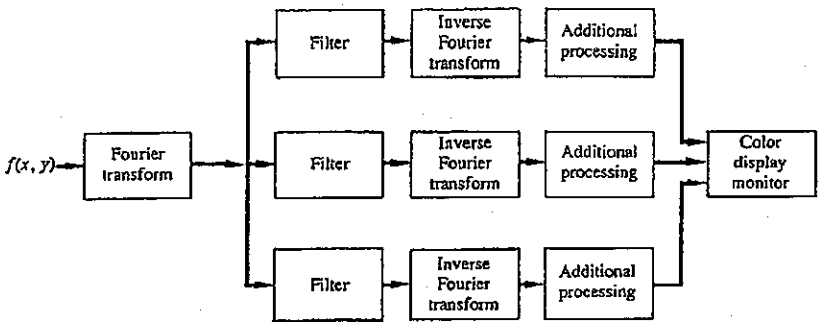


Figure 4.39 A filtering model for pseudo-color image enhancement.

where

$$D_1(u, v) = \{(u - u_0)^2 + (v - v_0)^2\}^{1/2} \tag{4.7-9}$$

and

$$D_2(u, v) = \{(u + u_0)^2 + (v + v_0)^2\}^{1/2} \tag{4.7-10}$$

The procedure can be extended in a similar manner to four or more regions. The Butterworth filter given in Section 4.4.2 can also be applied directly to band rejection by following the technique just described for the ideal filter. Figure 4.40 shows a perspective plot of a typical IBRF transfer function.

The filter discussed above is localized about some point off the origin of the Fourier transform. If it is desired to remove a band of frequencies centered about the origin, we can consider symmetric filters similar to the low and highpass filters discussed earlier. The procedure is illustrated for the ideal and Butterworth filters.

A radially symmetric ideal bandreject filter, which removes a band of frequencies about the origin, is given by the relation

$$H(u, v) = \begin{cases} 1 & \text{if } D(u, v) < D_0 - \dfrac{W}{2} \\ 0 & \text{if } D_0 - \dfrac{W}{2} \leq D(u, v) \leq D_0 + \dfrac{W}{2} \\ 1 & \text{if } D(u, v) > D_0 + \dfrac{W}{2}, \end{cases} \tag{4.7-11}$$

where W is the width of the band and D_0 is its radial center. As is the case with all radially symmetric filters, this filter can be completely specified by a cross section.

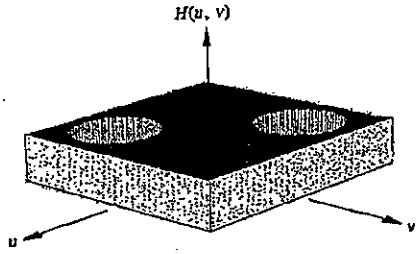


Figure 4.40 Ideal bandreject filter.

A radially symmetric Butterworth bandreject filter (BBRF) of order n has the transfer function

$$H(u, v) = \frac{1}{1 + \left[\dfrac{D(u, v)W}{D^2(u, v) - D_0^2}\right]^{2n}}, \tag{4.7-12}$$

where W is defined as the "width" of the band and D_0 is its center.

Bandpass filters pass frequencies in a specified band or region, while attenuating or completely suppressing, all other frequencies. Therefore they are exactly the opposite of bandreject filters. It then follows that, if $H_R(u, v)$ is the transfer function of any of the bandreject filters just discussed, the corresponding bandpass function $H(u, v)$ can be obtained simply by "flipping" $H_R(u, v)$; that is,

$$H(u, v) = -[H_R(u, v) - 1]. \tag{4.7-13}$$

Example: Plate VII(a) shows a monochrome image and Plates VII(b) and (c) show the results of using Butterworth filters. Plate VII(b) shows (in the red gun of a color monitor) the result of applying a highpass filter with the cut-off point at the circle enclosing 90% of the image energy (see Section 4.3.3). Plate VII(c) shows the highpass filtered image on the red gun, as well as a lowpass (blue gun) and bandpass (green gun) filtered version of Plate VII(a). The lowpass image was obtained with the cut-off point at the circle enclosing 98% of the image energy; the bandpass range was between the circles enclosing 20% and 95% of the energy. The principal enhancement resulting from this process was the increased visibility of the outer ring, which is almost invisible in the original image.

4.8 CONCLUDING REMARKS

The material presented in this chapter is representative of techniques commonly used in practice for digital image enhancement. It must be kept in mind, however, that this area of image processing is a dynamic field where reports of new techniques and applications are commonplace in the literature. For this reason, the topics included in this chapter were selected mostly for their value as fundamental material that would serve as a foundation for further study of this field.

REFERENCES

Additional reading for the material in Section 4.1 may be found in Gonzalez [1986]. Our discussion on histogram processing techniques is based on the papers by Hall *et al.* [1971], Hall [1974], Hummel [1974], Gonzalez and Fittes [1977], and Woods and Gonzalez [1981]. For further details on local enhancement see Ketcham [1976], Harris [1977], and Narendra and Fitch [1981].

The neighborhood averaging approach introduced in Section 4.3.1 is based on a similar discussion by Rosenfeld and Kak [1982]. For details on implementing median filters see Huang *et al.* [1979], Wolfe and Mannos [1979], and Chaudhuri [1983]. The lowpass-filtering concepts developed in Section 4.3.3 are based on a direct extension of one-dimensional filters where, instead of using a single variable, we used the distance from the origin of the Fourier transform in order to obtain circularly symmetric filter functions. This is also true of the other filters discussed in this chapter. For a discussion of one-dimensional filters see, for example, the books by Weinberg [1962] and by Budak [1974]. The method of smoothing by image averaging was first proposed by Kohler and Howell [1963].

Early references on image sharpening by differentiation are Goldmard and Hollywood [1951] and Kovasznay and Joseph [1953, 1955]. The Roberts gradient approach was proposed by Roberts [1965]. A survey of techniques used in this area a decade later is given by Davis [1975]. The articles by Prewitt [1970] and Frei and Chen [1977] are also of interest. More recent work in this field emphasizes computational speed, as exemplified by Lee [1983] and Chaudhuri [1983]. Our discussion on high-frequency emphasis is from Hall *et al.* [1971].

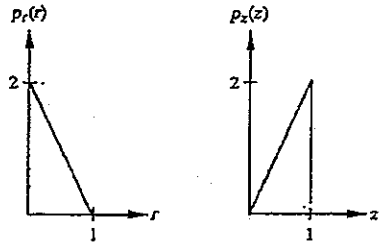
The material in Section 4.5 is based on a paper by Stockham [1972]; see also the book by Oppenheim and Schafer [1975]. The material in Section 4.6 is from Schutten and Vermeij [1980] and Meyer and Gonzalez [1983]. Basic material on color fundamendals may be found in the books by Walsh [1958] and by Kiver [1965]. The techniques discussed in Sections 4.6.2 through 4.6.4 are based on the papers by Smith [1963], Roth [1968], Billingsley *et al.* [1970], and Andrews *et al.* [1972]. Additional reading on pseudo-color image processing may be found in Green [1983].

PROBLEMS

4.1 Explain why the discrete histogram-equalization technique will not, in general, yield a flat histogram.

4.2 Suppose that a digital image is histogram-equalized. Show that a second pass of histogram equalization will produce exactly the same result as the first pass.

4.3 A given image has the gray level *PDF* $p_r(r)$ shown below. It is desired to transform the gray levels of this image so that they will have the specified $p_z(z)$ shown below. Assuming



continuous quantities, find the transformation (in terms of r and z) that will accomplish this.

4.4 Propose a method for updating the local histogram for use in the local enhancement technique discussed in Section 4.2.4.

4.5 Consider the neighborhood-averaging expression given in Eq. (4.3-1), and suppose that $M = 4$, corresponding to forming an average of the four immediate neighbors of (x, y), but excluding (x, y) itself.
a) Find the equivalent filter $H(u, v)$ in the frequency domain.
b) Show that this is a lowpass filter.

4.6 a) Develop a procedure for computing the median of an $n \times n$ neighborhood.
b) Propose a technique for updating the median as the center of the neighborhood is moved from pixel to pixel.

4.7 Under what condition does the Butterworth lowpass filter given in Eq. (4.3-6) become an ideal lowpass filter?

4.8 In a given industrial application, it is desired to use x-ray imaging to inspect the inside of certain iron castings. The objective is to look for voids in the castings, which typically appear as small blobs in the image. However, high noise content often makes inspection difficult, so it is decided to use image averaging to reduce the noise and thus improve visible contrast. The goal is to keep averaging to a minimum in order to reduce the time the parts have to remain stationary during imaging. After numerous experiments, it is concluded that decreasing the gray-level variance by a factor of 10 is sufficient. If the imaging device can produce 30 frames/sec, how long would the castings have to remain stationary during imaging to achieve the desired decrease in variance? Assume that the noise is uncorrelated and has zero mean.

4.9 The basic approach used to compute the digital gradient (Section 4.4.1) involves taking differences of the form $f(x, y) - f(x + 1, y)$.
a) Obtain the filter transfer function, $H(u, v)$, for performing the equivalent process in the frequency domain.
b) Show that this is a highpass filter.

4.10 A popular procedure for image enhancement combines high-frequency emphasis and histogram equalization to achieve edge sharpening and contrast enhancement.
a) Prove whether or not it matters which process is applied first.
b) If the order does matter, give a rationale for using one or the other method first.

4.11 Suppose that we are given a set of images generated by an experiment dealing with the analysis of stellar events. Each image contains a set of bright, widely scattered dots corresponding to stars in a sparse section of the universe. The problem is that the stars are barely visible, due to superimposed illumination resulting from atmospheric dispersion. If we model these images as the product of a constant illumination component with a set of impulses, give an enhancement procedure based on homomorphic filtering designed to bring out the image components due to the stars themselves.

4.12 With reference to the discussion in Section 4.6, show that if $H(u, v)$ is real and symmetric, $h(x, y)$ must also be real and symmetric.

4.13 In an automatic assembly application, it is desired to color code three classes of parts

simplify detection. However, only a monochrome TV camera is available. Propose a technique for using this camera to detect the three different colors.

4.14 A skilled medical technician is charged with the job of inspecting a certain class of images generated by an electron microscope. In order to simplify the inspection task, the technician decides to use digital image enhancement and, to this end, examines a set of representative images and finds the following problems: (1) bright, isolated dots that are of no interest; (2) lack of sharpness; (3) not enough contrast in some images; and (4) shifts in the average gray-level value, when this value should be K to perform correctly certain intensity measurements. The technician wishes to correct these problems and then color in constant red all gray levels in the band between I_1 and I_2, while keeping normal tonality in the remaining gray levels. Propose a sequence of processing steps that the technician can follow to achieve the desired goal.

TAB B

United States Patent [19]

Richard

[11] **Patent Number: 4,654,710**

[45] **Date of Patent: Mar. 31, 1987**

[54] **CONTRAST AMPLIFIER FOR VIDEO IMAGES**

[75] Inventor: **Christian J. Richard, Noyal Sur Vilaine, France**

[73] Assignee: **Thomson CSF, Paris, France**

[21] Appl. No.: **815,932**

[22] Filed: **Jan. 3, 1986**

[30] **Foreign Application Priority Data**

Apr. 1, 1985 [FR] France 85 00105

[51] Int. Cl.⁴ H04N 5/57

[52] U.S. Cl. 358/169; 358/39; 358/160; 364/734

[58] Field of Search 358/169, 39, 160, 32, 358/164, 184, 166, 139; 364/734, 715

[56] **References Cited**

U.S. PATENT DOCUMENTS

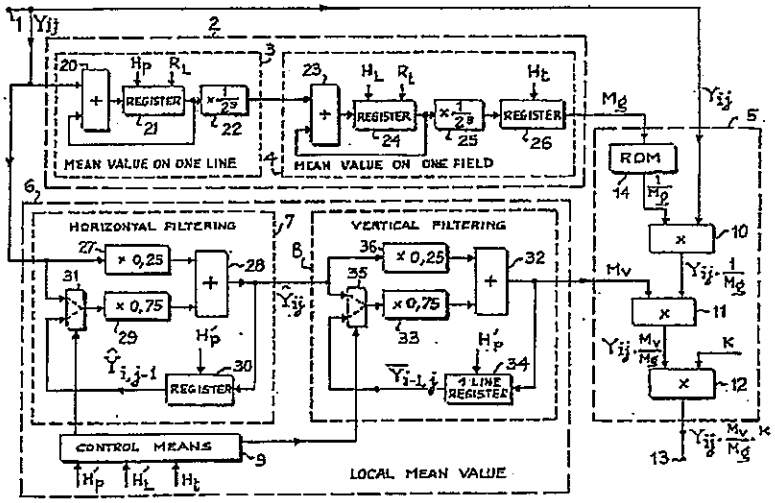
3,638,001	1/1972	Gordon	364/734 X
4,231,065	10/1980	Fitch et al.	358/166
4,553,619	11/1985	Fujinaga	364/734 X
4,603,353	7/1986	Henson	358/39 X
4,606,009	8/1986	Wiesmann	364/734 X

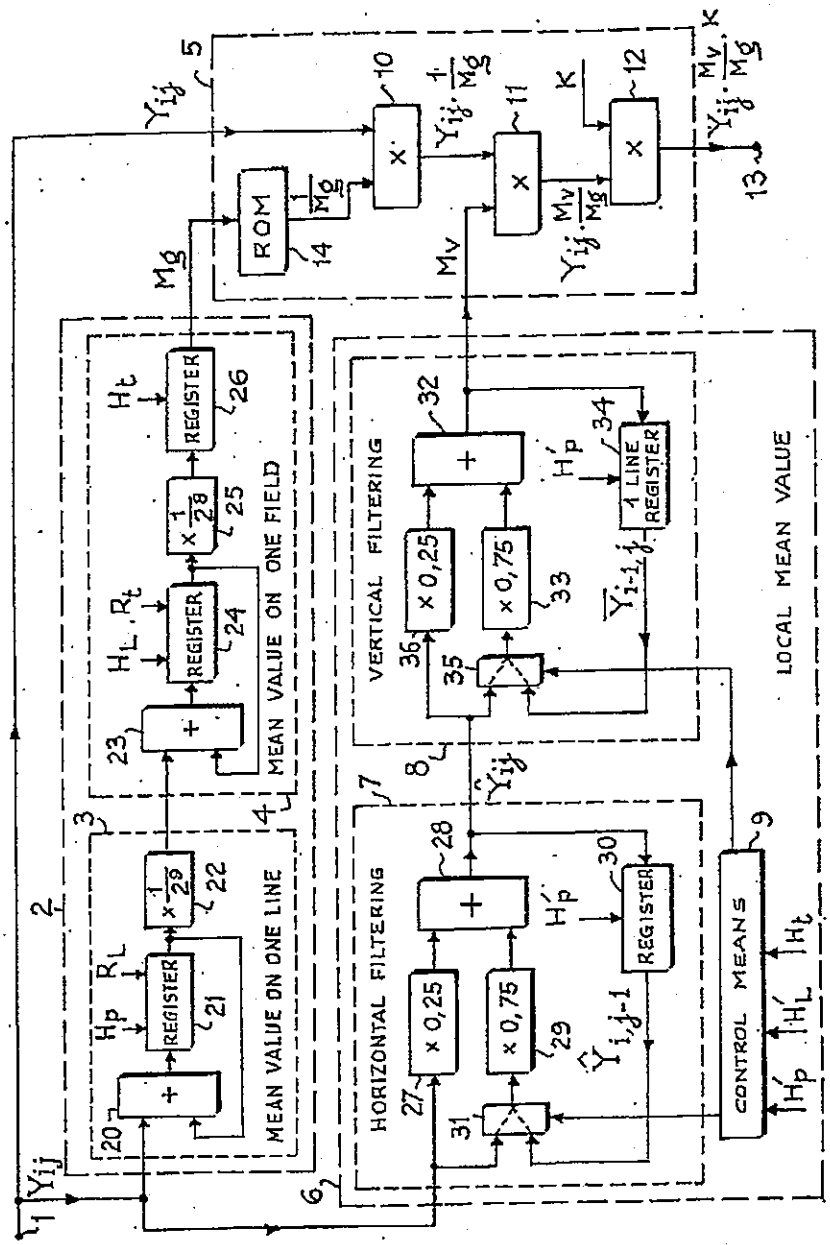
Primary Examiner—Michael A. Masinick
Assistant Examiner—E. Anne Toth
Attorney, Agent, or Firm—Cushman, Darby & Cushman

[57] **ABSTRACT**

A contrast amplifier for video images comprises means for computing a mean value M_g of luminance of all the points of a field, means consisting of a horizontal filter and a vertical filter for computing a local mean value M_v of luminance in the vicinity of a point being processed and means for multiplying the value of luminance of the point during processing by a variable coefficient which is proportional to the ratio M_v/M_g.

3 Claims, 1 Drawing Figure




MEAN VALUE ON ONE LINE
MEAN VALUE ON ONE FIELD
HORIZONTAL FILTERING
VERTICAL FILTERING
LOCAL MEAN VALUE
CONTROL MEANS
REGISTER
4 LINE REGISTER
ROM
× 0,25
× 0,75

CONTRAST AMPLIFIER FOR VIDEO IMAGES

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to devices which are known as contrast amplifiers and are employed for improving the visual perception of video images when these images have low contrast for such reasons as insufficient illumination, for example.

2. Description of the Prior Art

It is a known practice to enhance or amplify the contrast of video images by increasing the gain of the channel used for transmitting the luminance signal representing these images. This increase in gain is either uniform for all values of luminance or variable as a function of the luminance value. The disadvantage of this method is that it also increases the contrast of noise interference. Interference of this type becomes more troublesome as the contrast amplification is greater.

Another known method consists in equalizing the distribution of luminance values of each field throughout the range of possible values. To this end, the distribution of luminance values of a field is determined and each luminance value of the following field is corrected as a function of the distribution which is found in the case of the luminance values of the preceding field. French patent Application No. 2,456,448 filed by the present Applicant describes a method of this type. This known method has the advantage of putting the scale of luminance values to the best possible use but has the effect of increasing noise interference and is complex to apply in practice.

The contrast amplifier in accordance with the invention does not have the disadvantage of enhancing noise interference and is at the same time simple to construct. It is therefore particularly advantageous for processing images which are highly affected by noise. The amplifier makes advantageous use of the fact that noise affects only isolated points or a small number of interrelated points. The amplifier in accordance with the invention does not amplify the visibility of these points since it controls the gain of the channel which transmits the luminance signal, in a different manner at each image point, as a function of the luminance in the vicinity of this point and of the luminance of the entire image. The luminance in the vicinity of each point is determined by means of a so-called bidirectional spatial filtering process which consists of two recursive digital filtering operations performed successively. The mean luminance of an image can be estimated by considering the mean luminance of the preceding image in the case of a sequence of video images.

SUMMARY OF THE INVENTION

In accordance with the invention, a contrast amplifier for video images, each video image being represented by a sequence of digital values of luminance of the points of said image, comprises:

means for estimating a mean value M_g of luminance of all the points of each image in succession;

means for computing a local mean value M_v of luminance in the vicinity of a point being processed;

means for multiplying the value of luminance of the point being processed, by a variable coefficient which is proportional to the ratio M_v/M_g, the value thus obtained being such as to constitute the value of luminance of a point having enhanced contrast.

BRIEF DESCRIPTION OF THE DRAWINGS

Other features of the invention will be more apparent upon consideration of the following description and accompanying drawings in which the single FIGURE is a block diagram of an embodiment of the contrast amplifier according to the invention, processing a sequence of conventional video images. Each video image is composed of two interlaced fields which are processed as two independent images.

DESCRIPTION OF A PREFERRED EMBODIMENT

The embodiment considered by way of example comprises an input terminal 1, means 2 for computing a mean value M_g of luminance of all the points of a field, means 6 for computing a local mean value M_v of luminance of points which are adjacent to a point being processed for contrast enhancement, means 5 for multiplying the value of luminance of said point by a variable coefficient which is proportional to the ratio M_v/M_g and an output terminal 13 for delivering a sequence of numerical values of luminance having enhanced contrast.

The input terminal 1 receives a sequence of numerical values Y_{ij} of luminance of the points of a field and transmits them to an input of the means 2, to an input of the means 6 and to a first input of the means 5. The means 2 has an output connected to a second input of the means 5 for delivering the mean value M_g to said means 5. The means 6 have an output connected to a third input of the means 5 for delivering the mean value M_v to these latter.

The means 2 are constituted by a first computing device 3 for computing a mean value of luminance on each line and by a second computing device 4 for computing a mean value of luminance on a field by computing the average of the values supplied by the first device 3 during one field. The computing device 3 has an input which constitutes the input of the means 2 and has an output connected to an input of the second computing device 4. An output of the second computing device 4 constitutes the output of the means 2.

The first computing device 3 comprises an adder 20, a single-stage register 21 and a device 22 for dividing by 2^9. The input of the device 3 is connected to a first input of the adder 20. An output of the adder 20 is connected to a data input of the register 21. An output of the register 21 is connected to a second input of the adder 20 and to an input of the device 22. The adder 20 and the register 21 constitute an accumulator. The register 21 receives a clock signal H_p at the point-scanning frequency and receives a zero-reset signal R_L at the line-scanning frequency. The device 22 for dividing by 2^9 is constituted by special wiring for shifting the bits delivered by the register 21 by nine positions. These shifted bits are transmitted to the input of the second computing device 4.

The second computing device 4 comprises an adder 23, a single-stage register 24, a device 25 for dividing by 2^8 and a single-stage register 26. The input of the computing device 4 is connected to a first input of the adder 23. An output of the adder 23 is connected to a data input of the register 24. An output of the register 24 is connected to a second input of the adder 23 and to an input of the device 25. An output of the device 25 is connected to a data input of the register 26 and an out-

put of this latter constitutes the output of the device 4 and the output of the means 2.

The adder 23 and the register 24 constitute an accumulator. The register 24 receives a clock signal H_L at the line frequency and receives a zero-reset signal R_t at the field frequency. The device 25 for dividing by 2^8 is constituted by a special wiring which has the effect of shifting the bits delivered by the register 24 by eight positions before transmitting them to the input of the register 26. The register 26 has a clock input for receiving a clock signal H_t at the field frequency in order to store a mean value of luminance during an entire field period. The mean luminance value M_g computed during a field period is employed for amplifying (enhancing) the contrast during the following field period.

The means 6 for computing a local mean value M_v of luminance in the vicinity of a point being processed are constituted by a horizontal-filtering device 7 connected in series with a vertical-filtering device 8 and by control means 9 for initialization. The filtering devices 7 and 8 are recursive digital filtering devices. The so-called bidirectional spatial filtering achieved by a combination of these devices is comparable with the filtering which would be obtained by computing the arithmetical mean of the luminance values in a sliding "window" centered on the current point. The advantage of this combination, however, lies in the fact that it is conducive to greater simplicity of computations since the horizontal filtering and vertical filtering operations mentioned above are performed by means of two recursive computations which require a smaller number of calculations and less storage of values.

Computation of a mean value of luminance for each line is performed on 512 points, that is to say on a number of points slightly smaller than the number of real points in a line. The number 512 makes it possible to simplify the device 3 since it is very easy to perform a division by 2^9. Computation of the mean value of luminance of the points of a field is carried out on 256 lines, which is smaller than the real number of lines in a field according to European standards since it is an easy matter to perform a division by 2^8. The signal generators for delivering the clock signals H_l, R_L, R_t, H_t are not shown in the FIGURE since their construction is within the capacity of any one versed in the art. The clock signal H_p delivers only 512 pulses per line and the clock signal H_L delivers only 256 pulses per field.

The horizontal filtering device 7 computes, in respect of the j^{th} point on the i^{th} line, a horizontally-filtered luminance value $\hat{Y}_{ij}$ in accordance with the formula:

$$\hat{Y}_{ij}=0.25.Y_{ij}+0.75.\hat{Y}_{ij-1}$$

where $\hat{Y}_{ij-1}$ is a value filtered by the filtering device 7 and computed in respect of the immediately preceding point.

Said device 7 comprises a device 27 for multiplication by 0.25, a device 29 for multiplication by 0.75, an adder 28, a single-stage register 30 and a multiplexer 31. The multiplication devices 27 and 29 are constituted by read-only memories (ROMs). The input of the means 6 is connected to an input of the device 7 and this latter is connected to an input of the device 27 for multiplication by 0.25 and to a first input of the multiplexer 31. A second input of the multiplexer 31 is connected to an output of the register 30 and an output of the multiplexer 31 is connected to an input of the device 29 for multiplication by 0.75. A control input of the multiplexer 31 is connected to an output of the control means 9 for initialization.

An output of the device 27 is connected to a first input of the adder 28. An output of the adder 28 is connected to the output of the device 7 and to an input of the register 30. An output of the register 30 delivers the value $\hat{Y}_{ij-1}$ to the second input of the multiplexer 31. An output of the device 29 is connected to a second input of the adder 28. The register 30 has a clock input for receiving a clock signal H'_p at the point-scanning frequency and the number of pulses of which corresponds to the real number of points in each line. The generator for the clock signal H'_p is not shown in the FIGURE and construction of this generator is within the capacity of those versed in the art.

The vertical-filtering device 8 has a structure which is similar to that of the horizontalfiltering device 7 but comprises a register 34 for producing a delay which corresponds to one line instead of the register 30 which produces a delay corresponding to one image point. The register 34 delivers a luminance value $\hat{Y}_{i-1,j}$ corresponding to the point equivalent to the point being processed and located on the immediately preceding line, this value being filtered by the devices 7 and 8. The device 8 comprises a device 36 for multiplication by 0.25, a device 33 for multiplication by 0.75, an adder 32, the register 34 and a multiplexer 35.

An input of the device 8 is connected to the output of the device 7 and is connected to an input of the device 36 for multiplication by 0.25 and to a first input of the multiplexer 35. A second input of the multiplexer 35 is connected to the output of the register 34. An output of the multiplexer 35 is connected to an input of the device 33 for multiplication by 0.75. A control input of the multiplexer 35 is connected to an output of the means 9.

An output of the device 36 is connected to a first input of the adder 32. The output of the adder 32 is connected to the output of the device 8, this output being such as to constitute the output of the means 6, and to an input of the register 34. The output of the register 34 delivers the value $\hat{Y}_{i-1,j}$ to the second input of the multiplexer 35. An output of the device 33 is connected to a second input of the adder 32. The register 34 has a control input for receiving the clock signal H'_p.

Since the horizontal-filtering device 7 and the vertical-filtering device 8 are connected in series, they perform a spatial filtering operation which produces a filtered value M_v. This value is similar to the mean value which would be obtained by computing the arithmetical mean of the luminance values of the points located in a sliding window centered on the point being processed. The recursive filtering operation carried out by the device 7 is not possible for the first point of each line and the vertical recursive filtering operation performed by the device 8 is not possible for the first line. It is for this reason that means 9 have been provided for initialization.

The means 9 receive the clock signals H'_p and H_t and a clock signal H'_L. The clock signal H'_L delivers at the line-scanning frequency a number of pulses equal to the real number of lines of each field.

During the period which corresponds to processing of the first point of each line, the means 9 deliver a signal for controlling the multiplexer 31 in such a manner as to ensure that this latter connects the input of the device 29 to the input of the device 7. Thus the filtered value $\hat{Y}_{ij-1}$ of luminance of the non-existent preceding

point is replaced by the non-filtered value Y_{ij} of luminance of the point being processed. The device 7 therefore computes: $0.25.Y_{ij}+0.75.Y_{ij}=Y_{ij}$. In the case of the first point of each line, the filtered value is therefore strictly identical with the original value. During each period corresponding to processing of the other points of each line, the means 9 deliver a signal for controlling the multiplexer 31 in such a manner as to ensure that this latter connects the input of the device 29 to the output of the register 30. The device 7 then computes a filtered value in accordance with the recurrence formula :

$$\hat{Y}_{ij}=0.25.Y_{ij}+0.75.\hat{Y}_{i,j-1}$$

During the period which corresponds to processing of all the points of the first line of each field, the means 9 deliver a signal for controlling the multiplexer 35 in such a manner as to ensure that this latter connects the input of the device 33 to the input of the device 8. Thus the filtered value $\bar{Y}_{i-1,j}$ of the point equivalent to the current point on the non-existent preceding line is replaced by the horizontally filtered value $\hat{Y}_{ij}$ of the current point. The device 8 therefore computes: $0.25.\hat{Y}_{ij}+0.75.\hat{Y}_{ij}=\bar{Y}_{ij}$.

In the case of the points of the first line, spatial filtering is therefore reduced to horizontal filtering. During the time required to process the other lines of each field, the means 9 deliver a signal for controlling the multiplexer 35 in such a manner as to ensure that this latter connects the output of the register 34 to the input of the device 33. The device 8 then computes a filtered value in accordance with the recurrence formula:

$$\bar{Y}_{ij}=0.25.\hat{Y}_{ij}+0.75.\bar{Y}_{i-1,j}$$

This filtered value $\bar{Y}_{ij}$ constitutes the mean value M_v of luminance in the vicinity of the point considered.

The means 5 for multiplying the value Y_{ij} of luminance of the point being processed by a variable coefficient which is proportional to the ratio M_v/M_g are constituted by a read-only memory (ROM) 14 and three digital multipliers 10, 11 and 12. An address input of the read-only memory 14 is connected to the second input of the means 5 for receiving the mean value M_g. An output of the memory 14 delivers a value $1/M_g$ to a first input of the multiplier 10. A second input of the multiplier 10 is connected to the first input of the means 5 for receiving a non-filtered luminance value Y_{ij}. An output of the multiplier 10 is connected to a first input of the multiplier 11 in order to deliver a value $Y_{ij}.1/M_g$ to this latter.

A second input of the multiplier 11 is connected to the third input of the means 5 for receiving the value M_v of the local mean. An output of the multiplier 11 is connected to a first input of the multiplier 12 in order to deliver a value $Y_{ij}.M_v/M_g$ to said first input. A second input of the multiplier 12 receives a constant value K which can be adjusted by an operator in order to adjust the contrast at will. An output of the multiplier 12 constitutes the output of the means 5 and is connected to the output terminal 13 of the contrast amplifier. Said output delivers a value $Y_{ij}.M_v/M_g.K$.

The coefficient M_v/M_g can be lower than or higher than 1 according to whether the vicinity of the point considered has a mean luminance value which is lower than or higher than the mean luminance value of the preceding field. Depending on which of these two cases applies, the effect of the contrast amplifier is therefore to reduce the luminance of the current point in order to bring it close to the value of black or respectively to increase said luminance in order to bring it close to the value of pure white.

Granular noise is represented by isolated grey points which are particularly visible in dark areas. As a result of bidirectional spatial filtering, the luminance values affected by noise produce very little disturbance in computation of the local mean value M_v and are consequently treated in the same manner as the values of the adjacent points. In the case of areas which are darker than the general mean value, these areas have an even darker appearance after processing including the points affected by noise. This is the contrary to what may be observed in the conventional method which consists simply in producing a uniform increase in gain in respect of all luminance values and therefore has the effect of increasing the luminosity of isolated grey points in the dark areas.

The invention is not limited to the example of construction described in the foregoing and many alternative embodiments are within the competence of those versed in the art. It is possible in particular to change the order of the horizontal filtering device 7 and of the vertical filtering device 8 or to change the order of the two multiplication operations which are necessary for computing $Y_{ij}.M_v/M_g.K$.

It is also possible to replace the means 6 by any other known device for computing a mean value of luminance in the vicinity of the point being processed.

The invention can be applied in particular to television pictures having low levels of illumination.

What is claimed is:

1. A contrast amplifier for video images, each video image being represented by the sequence of digital values of luminance of a points of said image, comprising:

first means for estimating a mean value M_g of luminance of all the points of each image in succession;

second means for computing a local mean value M_v of luminance in the vicinity of a point being processed;

means for multiplying the value of luminance of the point being processed by a variable coefficient which is proportional to the ratio M_v/M_g, the value thus obtained being such as to constitute the value of luminance of a point having enhanced contrast.

2. A contrast amplifier according to claim 1, wherein the means for computing a local mean value M_v of luminance in the vicinity of a point comprise:

first recursive filtering means for receiving the sequence of luminance values X in order to compute a filtered value $\bar{X}$ in accordance with the formula:

$$\bar{X}=\alpha_1.X+(1-\alpha_1).A \text{ with } 0\leq\alpha_1\leq 1$$

where A is a filtered luminance value which has previously been computed by the first means in respect of one of the two points which are immediately adjacent to the point considered, namely the point which precedes said point on the same line and the corresponding point on the line scanned immediately beforehand;

and second recursive filtering means in series with the first means for computing the value M_v in accordance with the formula:

$$M_v=\alpha_2.\bar{X}+(1-\alpha_2).B \text{ with } 0\leq\alpha_2\leq 1$$

where B is a filtered luminance value which has previously been computed by the first and second means in respect of the other of the two points immediately adjacent to the point considered.

3. A contrast amplifier according to claim 2, wherein

the means for estimating a mean value M_e of luminance of all the points of each image comprise:

a first device for computing a mean value of luminance of the points on each line of an image,

a second device for computing a mean value of all the mean values computed by the first device in the case of each image.

* * * * *

TAB C

REFERENCES

[1] S. Watanabe, *Cognition and Pattern* (in Japanese). Tokyo, Japan: Iwanami, 1978.

[2] H. Bergson, *Evolution Creatrice*. Paris, France: Felix Alcan, 1907.

[3] S. Fujiwhara, *Jap. J. Astron. Geophys.*, vol. 5, p. 143, 1923; *J. Roy. Met. Soc.*, vol. 49, p. 89, 1923; see also S. Fujiwhara, *Kumo wo Tsukamu Hanashi* (in Japanese). Tokyo, Japan: Iwanami, 1940.

[4] N. Wiener, *Cybernetics*, 2nd ed. Cambridge, MA: MIT Press, 1962; see also *The Human Use of Human Being*, 2nd ed. Boston, MA: Houghton Mifflin, 1956.

[5] S. Watanabe, "The cybernetical view of time," in *Progress in Biocybernetics*, vol. 3, Wiener and Schade, Eds. Amsterdam, The Netherlands: Elsevier, 1966; see also "Norbert Wiener and cybernetical concept of time," *IEEE Trans. Syst., Man, Cybern.*, p. 392, May 1975.

[6] P. Glansdorff and I. Prigogine, *Thermodynamical Theory of Structure, Stability and Fluctuation*. New York: Wiley, 1971.

[7] S. Watanabe, "Information-theoretical aspects of inductive and deductive inference," *IBM J. Res. Develop.*, vol. 4, p. 208, 1960.

[8] ——, "Learning process and inverse H-theories," *IRE Trans. Inform. Theory*, vol. IT-8, p. 248, 1962.

[9] ——, *Knowing and Guessing*. New York: Wiley, 1969, p. 287.

[10] ——, *Knowing and Guessing*. New York: Wiley, quoted in footnote 9, p. 395; "Karhunen-Loève expansion and factor analysis," in *Trans. 4th Prague Conf. on Inform. Theory, etc.*, 1965. Prague: Czechoslovak Acad. Sci., 1969.

[11] S. Watanabe and E. T. Harada, "A dynamical model of clustering," in *Proc. 2nd Int. Joint Conf. on Pattern Recognition*. Copenhagen, 1974, p. 413; S. Watanabe, "Further report on coalescence model of clustering," in *Proc. 3rd Int. Joint Conf. on Pattern Recognition*. San Diego, 1976, p. 176; "Application of dynamical coalescence model to pattern recognition," in *Proc. 4th Int. Joint Conf. on Pattern Recognition*. Kyoto, 1978.

[12] S. Watanabe, *Knowing and Guessing*. New York: Wiley, 1969, pp. 14, 17.

[13] A. Rosenfeld, R. A. Hummel, and S. W. Zucker, "Scene labeling by relaxation operations," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-6, no. 6, p. 420, 1976.

Digital Image Enhancement and Noise Filtering by Use of Local Statistics

JONG-SEN LEE

Abstract—Computational techniques involving contrast enhancement and noise filtering on two-dimensional image arrays are developed based on their local mean and variance. These algorithms are nonrecursive and do not require the use of any kind of transform. They share the same characteristics in that each pixel is processed independently. Consequently, this approach has an obvious advantage when used in real-time digital image processing applications and where a parallel processor can be used. For both the additive and multiplicative cases, the *a priori* mean and variance of each pixel is derived from its local mean and variance. Then, the minimum mean-square error estimator in its simplest form is applied to obtain the noise filtering algorithms. For multiplicative noise, a statistical optimal linear approximation is made. Experimental results show that such an assumption yields a very effective filtering algorithm. Examples on images containing 256 × 256 pixels are given. Results show that in most cases the techniques developed in this paper are readily adaptable to real-time image processing.

Index Terms—Digital image enhancement, local statistics, noise filtering, real-time processing.

Manuscript received December 16, 1977; revised April 29, 1979.

The author is with the Naval Research Laboratory, Washington, DC 20375.

INTRODUCTION

Image processing on digital computers has been gaining in acceptance in recent years [1]–[3]. Early techniques in image processing concentrated mostly on procedures that were carried out computationally in the frequency domain (Fourier or Walsh), which was a natural extention of one-dimensional linear signal-processing theory. In due course it became well-known that computing a two-dimensional transform for a large data array is a very time-consuming activity even with fast transform techniques on large computers. Hence implementation of frequency domain procedures for real-time processing of images appears less than promising. More recent works based on an application of Kalman filtering algorithms [4] or Bayesian estimation extended to two-dimensional arrays led to the concept of a recursive filtering algorithm [5], [6]. The power of recursive algorithms for real-time one-dimensional signal processing are well established. However, when applied to a two-dimensional array, the algorithm operates in the spatial domain in which pixels have to be processed sequentially. As a consequence, the procedure is no longer computationally efficient and loses its attraction for real-time processing.

Algorithms developed in this paper share a particular characteristic in that each pixel can be processed separately without waiting for its neighboring pixels to be processed. This characteristic permits a direct implementation of these algorithms for real-time image processing. Applying local statistics to image processing is not a new idea. Ketcham *et al.* [7] used the entire local histogram for real-time image enhancement, and Wallis [8] applied local mean and variance to filter out scan line noise with striking results. This paper extends this family of algorithms to contrast enhancement and noise filtering. Both additive white noise and multiplicative white noise cases are considered. Most additive noise filtering approaches utilize the fast Fourier transform, convolution, or recursive algorithms. In the transform and convolution methods, the autocorrelation between pixels is either assumed or approximated, and in the recursive algorithm, a linear causal or semi-causal autoregressive image model is generally assumed. The techniques based on the use of local mean and variance described in this paper deviate from these approaches. The basic assumption is that the sample mean and variance of a pixel is equal to the local mean and variance of all pixels within a fixed range surrounding it. The validity of this assumption is debatable, but so are most other statistical image representations encountered in the current practice. In the additive noise filtering case, the *a priori* mean (variance) of the estimated image is calculated as the difference between the mean (variance) of the noise corrupted image and the mean (variance) of the noise by itself. This technique is extended to include multiplicative noise filtering and also the case involving both multiplicative and additive noise. Although this simple approach may not have the mathematical elegance and sophistication of a few other techniques, our experimental results and those reported by Wallis [8] indicate that the local mean and variance technique is a very effective tool in both contrast stretching and noise filtering applications.

Let x_{ij} be the brightness of the pixel (i, j) in a two-dimensional $N \times N$ image. The local mean and variance are calculated over a $(2n + 1) \times (2m + 1)$ window. The local mean is defined as

$$m_{i,j} = \frac{1}{(2n+1)(2m+1)} \sum_{k=i-n}^{i+n} \sum_{l=j-m}^{j+m} x_{k,l} \tag{1}$$

Similarly, the local variance is

$$v_{i,j} = \frac{1}{(2n+1)(2m+1)} \sum_{k=i-n}^{i+n} \sum_{l=j-m}^{j+m} (x_{k,l} - m_{i,j})^2 \tag{2}$$

In this paper, Section II is devoted to spatial contrast enhancement for which only the local mean is required. Section III deals with additive noise filtering. Section IV treats images corrupted by multiplicative noise, and Section V extends the results of Sections III and IV to images corrupted by both additive and multiplicative noise. In Section V, a simplified algorithm is discussed to approximate the local mean and variance and future research using local statistics is briefly mentioned.

II. Spatial Contrast Enhancement

Gray level rescaling, high-pass filtering, and histogram redistribution [9], are the most popular techniques in image contrast enhancement. Wallis [8] suggested an algorithm based on local mean and variance in which each pixel is required to have a "desirable" local mean m_d and a "desirable" local variance v_d such that

$$x_{i,j} = m_d + \sqrt{\frac{v_d}{v_{i,j}}}\,(x_{i,j} - m_{i,j}) \tag{3}$$

where, in (3), $m_{i,j}$ and $v_{i,j}$ are local mean and variance as given by (1) and (2). It is easy to verify that the $x_{i,j}$ has a mean m_d and variance v_d if we consider $m_{i,j}$ and $v_{i,j}$ as the true mean and variance of $x_{i,j}$. The main drawback of this technique is that it tends to enhance subtle details at the expense of the principal features which are lost in the process. Fig. 1 shows the original image and the image processed by (3). The river in the original image and other large objects are difficult to recognize in the processed image. To compensate for this, an algorithm is designed such that a pixel $x_{i,j}$ will maintain its local mean, and yet permits its variance to be modified by a factor of its original local variance. The modified algorithm is

$$x_{i,j} = m_{i,j} + k(x_{i,j} - m_{i,j}) \tag{4}$$

where k, the gain, is the ratio of new local standard deviation to the original standard deviation. This scheme has a distinct computational advantage over the scheme posed in (3). The computation of local variance $v_{i,j}$ is not required and only $m_{i,j}$ need be computed. If $k > 1$, the image will be sharpened as if acted upon by a high-pass filter. If $0 \leq k < 1$, the image will be smoothed as if passed through a low-pass filter. In the extreme case, $k = 0$ and $x_{i,j}$ is equal to its local mean $m_{i,j}$.

This algorithm can be easily extended to enhance images with an ill-distributed histogram. Let $g(x)$ be the gray level rescaling function [9], then (4) is modified to

$$x_{i,j} = g(m_{i,j}) + k(x_{i,j} - m_{i,j}) \tag{5}$$

In the case of a linear gray level stretch, $g(x)$ is a linear relation. Several images are processed and shown in Fig. 1 using $g(x) = ax + b$, where $a = 0.9$ and $b = 13$ to allow contrast enhancement at both ends of gray scale (between 0 and 255). The linear function $g(x)$ used in this picture yields an effective contrast stretch in both the highlights and the dark areas of the image. The window size of 3×3 or 5×5 is large enough to carry out contrast enhancement. For noise filtering (to be discussed in later sections), however, a 7×7 or higher dimensional window is more desirable but at the expense of image resolution. All images of Fig. 1 are processed by the use of a 5×5 window. Fig. 1(c) shows that for $k = 2$, both the gross features and subtle details are enhanced in the same proportion. The case $k = 1$ has no effect on the image. For $k = 0.5$, the image is blurred as if passed through a low-pass filter. For $k = 0$, the image is averaged over the 5×5 window.

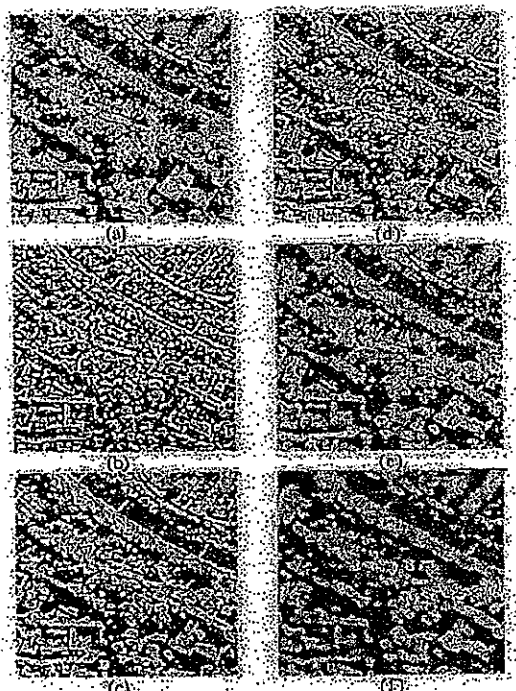


Fig. 1. (a) Original (or $k = 1$). (b) Wallis algorithm. (c) $k = 2$. (d) $k = 3$. (e) $k = 0.5$. (f) $k = 0$.

III. Additive Noise Filtering

In this section, attention is focused on filtering of images degraded by white additive noise. Most current approaches to this problem employ frequency domain techniques, direct inversion, or recursive Kalman filtering. In the present paper, we derive a very simple algorithm based on the use of local mean and variance of an image. Let $z_{i,j}$ be the degraded pixel $x_{i,j}$ then

$$z_{i,j} = x_{i,j} + w_{i,j} \tag{6}$$

where $w_{i,j}$ is a white random sequence with

$$E[w_{i,j}] = 0$$

and $E[w_{i,j} w_{k,l}] = \eta_w^2 \delta_{i,k} \delta_{j,l}$, where $\delta_{i,k}$ is the Kronecker delta function and E is the expectation operator. (The distribution of $w_{i,j}$ is irrelevant in this derivation.) The problem is to estimate $x_{i,j}$ given $z_{i,j}$ and the noise statistics.

From (6), we have

$$\bar{x}_{i,j} \triangleq E[x_{i,j}] = E[z_{i,j}] = \bar{z}_{i,j} \tag{7}$$

and

$$Q_{i,j} \triangleq E[(x_{i,j} - \bar{x}_{i,j})^2] = E[(z_{i,j} - \bar{z}_{i,j})^2] - \sigma_w^2 \tag{8}$$

In most filtering techniques, the *a priori* mean and variance of $x_{i,j}$ are derived from an assumed autocorrelation model or, recursively, from an autoregressive Markov sequence. In either method, it is an approximation. Assume that $\bar{x}_{i,j}$ and $Q_{i,j}$ are the *a priori* mean and variance of $x_{i,j}$, which in turn are approximated by the local mean and variance of (7) and (8). Under this assumption, it is very easy to obtain a filtering algorithm either by minimizing the mean-square error or

by weighted least-square estimation [10]. Both methods will produce the same algorithm. The estimated $\hat{x}_{i,j}$, $\hat{x}_{i,j}$ is computed by

$$\hat{x}_{i,j}=\bar{x}_{i,j}+k_{i,j}(z_{i,j}-\bar{x}_{i,j}) \tag{9}$$

where the gain is given by

$$k_{i,j}=\frac{Q_{i,j}}{Q_{i,j}+\sigma_1^2} \tag{10}$$

Equation (9) can be written as

$$\hat{x}_{i,j}=(1-k_{i,j})\bar{x}_{i,j}+k_{i,j}(z_{i,j}) \tag{11}$$

Since $Q_{i,j}$ and σ_1^2 are both positive, $k_{i,j}$ will lie between 0 and 1. A simple intuitive interpretation is that for a low signal-to-noise ratio region $Q_{i,j}$ is small compared with σ_1^2, $k_{i,j}\simeq 0$, and the estimated $\hat{x}_{i,j}$ is $\bar{x}_{i,j}$. Conversely, for a high signal-to-noise ratio region, $Q_{i,j}$ is much larger than σ_1^2, $k_{i,j}\simeq 1$, and $\hat{x}_{i,j}\simeq z_{i,j}$, the corrupted pixel. The use of different window sizes will greatly affect the quality of processed images. If the window is too small, the noise filtering algorithm is not effective. If the window is too large, subtle details of the image will be lost in the filtering process. Our experiments indicate that a 7×7 window is a fairly good choice. All images presented in this and later sections are processed by the 7×7 window.

Fig. 2 shows the original image, the image contaminated with additive noise and the estimated image produced by the local mean and variance algorithm. Clearly, in a smooth area, the pixel is averaged over the window and in a high contrast or edge area, the noise corrupted pixels are weighted higher than their local mean value. Fig. 2(d), (e), and (f) are the plots of intensity along a specific scan line for the original, the noise corrupted, and the processed images, respectively. This algorithm works equally well for an image corrupted by a Gaussian noise. Results for the latter case are shown in Fig. 3(a) and (b).

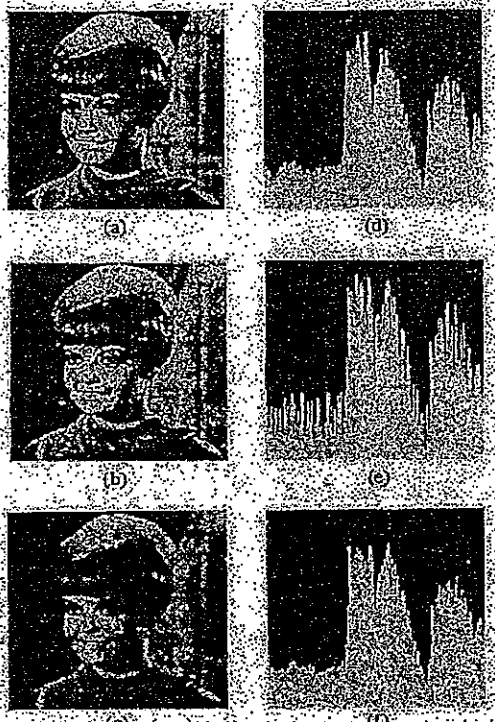


Fig. 2. (a) Original. (b) Original plus additive uniform noise (−30, 30). (c) Additive noise removed with (7×7) mesh, $\sigma_1^2=300$. (d) Original intensity profile along a scan line. (e) Profile at (d) contaminated by additive noise. (f) Profile at (e) filtered for noise.

Fig. 3. (a) Original plus additive Gaussian noise, $\sigma_1^2=300$. (b) Noise removed.

IV. Multiplicative Noise Filtering

Images containing multiplicative noise have the characteristic that the brighter the area the noisier it is. Mathematically, the degraded pixel can be represented by

$$z_{i,j}=x_{i,j}u_{i,j} \tag{12}$$

where $E[u_{i,j}]=\bar{u}_{i,j}$ and

$$E[(u_{i,j}-\bar{u}_{i,j})(u_{k,l}-\bar{u}_{k,l})]=\sigma_2^2\delta_{i,k}\delta_{j,l}.$$

Nahi and Naraghi [11] treat this problem via the Kalman–Bucy approach which necessitates solving nonlinear estimation problem by numerical integration. In this paper, the nonlinearity in (12) is treated differently. An optimal linear approximation of (12) is used to produce a filtering algorithm similar to that for the additive noise case. Experimental results show that the derived algorithm is a very promising one. Let

$$z'_{i,j}=A\bar{x}_{i,j}+Bu_{i,j}+C \tag{13}$$

where A, B, and C are nonrandom variables. They are to be chosen to minimize the mean square error between $z_{i,j}$ and $z'_{i,j}$ and also to make $z'_{i,j}$ an unbiased estimate of $z_{i,j}$. For $z'_{i,j}$ to be unbiased estimate of $z_{i,j}$, we must have

$$A\bar{x}_{i,j}+B\bar{u}_{i,j}+C=\bar{x}_{i,j}\bar{u}_{i,j}$$

or

$$C=\bar{x}_{i,j}\bar{u}_{i,j}-A\bar{x}_{i,j}-B\bar{u}_{i,j}. \tag{14}$$

Substituting (14) into (13) and forming the mean-square error, we arrive at the performance index to be minimized,

$$J=E[A(x_{i,j}-\bar{x}_{i,j})+B(u_{i,j}-\bar{u}_{i,j})-(x_{i,j}u_{i,j}-\bar{x}_{i,j}\bar{u}_{i,j})]^2.$$

Upon carrying out the necessary mathematical procedures, we obtain the following relation:

$$z'_{i,j}=\bar{u}_{i,j}\bar{x}_{i,j}+\bar{x}_{i,j}(u_{i,j}-\bar{u}_{i,j}). \tag{15}$$

It is not surprising to find that (15) actually is the first-order Taylor series expansion of $z_{i,j}$ about $(\bar{x}_{i,j},\bar{u}_{i,j})$.

The *a priori* mean and variance of $x_{i,j}$ are computed from (12) and are given by

$$\bar{x}_{i,j}=\bar{z}_{i,j}/\bar{u}_{i,j} \tag{16}$$

and

$$Q_{i,j}=\frac{\mathrm{var}(z_{i,j})+\bar{z}_{i,j}^2}{\sigma_2^2+\bar{u}_{i,j}^2}-\bar{x}_{i,j}^2 \tag{17}$$

in which $\mathrm{var}(z_{i,j})$ is the variance of $z_{i,j}$. The quantities $\bar{z}_{i,j}$ and $\mathrm{var}(z_{i,j})$ are approximated by the local mean and local variance of the corrupted image. Using (16) and (17), and

applying the Kalman filtering algorithm to (15), we have

$$\hat{x}_{i,j} = \bar{x}_{i,j} + k_{i,j}(z_{i,j} - u_{i,j}\bar{x}_{i,j}) \tag{18}$$

in which

$$k_{i,j} = \frac{u_{i,j}Q_{i,j}}{\bar{x}_{i,j}^2\sigma_2^2 + u_{i,j}^2 Q_{i,j}}. \tag{19}$$

An experimental example is shown in Fig. 4, in which the original image is corrupted by multiplicative noise uniformly distributed between 0.7 and 1.0, and the estimated image has been processed by the algorithm developed in this section. Considerable improvement is shown in the processed image, thus substantiating the effectiveness of the local mean and variance technique.

V. Filtering of Combined Additive and Multiplicative Noise

It is very easy to extend the algorithms of previous sections to deal with images corrupted by both additive and multiplicative noise. A noise-corrupted image is described by

$$z_{i,j} = x_{i,j}u_{i,j} + w_{i,j} \tag{20}$$

in which the statistical characteristics are the same as given in Sections III and IV. Assume that $u_{i,j}$ and $w_{i,j}$ are independent white noises. This independence assumption can be removed, but the result is a more complicated formulation. Following the idea of an optimal linear approximation of Section IV, we have

$$\hat{x}_{i,j} = \bar{u}_{i,j}\bar{x}_{i,j} + \bar{x}_{i,j}(u_{i,j} - \bar{u}_{i,j}) + w_{i,j}.$$

The formulas for the *a priori* mean and variance of $x_{i,j}$ of Section IV are modified to read

$$\bar{x}_{i,j} = (z_{i,j} - \bar{w}_{i,j})/\bar{u}_{i,j} \tag{21}$$

and

$$Q_{i,j} = \frac{\mathrm{var}(z_{i,j}) + \bar{z}_{i,j}^2}{\sigma_2^2 + \bar{u}_{i,j}^2} - \bar{x}_{i,j}^2 - \sigma_1^2.$$

The filtering algorithm is

$$\hat{x}_{i,j} = \bar{x}_{i,j} + k_{i,j}(z_{i,j} - \bar{u}_{i,j}\bar{x}_{i,j} - \bar{w}_{i,j}) \tag{22}$$

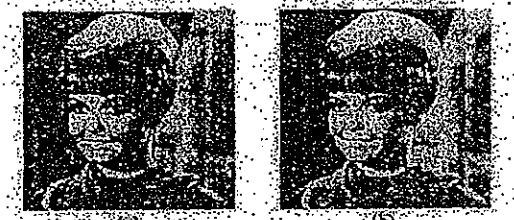
in which

$$k_{i,j} = \frac{\bar{u}_{i,j}Q_{i,j}}{\bar{x}_{i,j}^2\sigma_2^2 + \bar{u}_{i,j}^2 Q_{i,j} + \sigma_1^2}. \tag{23}$$

Fig. 5(a) shows the image corrupted by an additive noise uniformly distributed between gray levels −20 and +20 and also a multiplicative noise uniformly distributed between multiplicative factors 0.7 and 1.0. The processed image, Fig. 5(b) shows a very significant improvement over the original image.

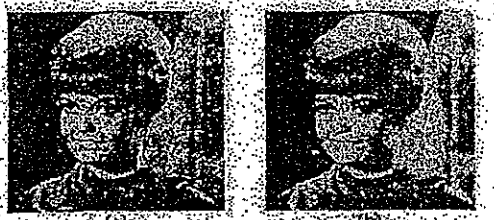
VII. Remarks and Conclusions

The principal computational load of the developed algorithms is in the calculation of the local means and variances, especially the latter. To lighten this burden Wallis [8] proposed a fast algorithm in which the image is partitioned into square subregions over which the local mean and variance are computed. Then the local mean and variance of a particular pixel are approximated by the use of two-dimensional interpolation formulas. Results, as reported by Wallis [8], are quite favorable. It is believed that Wallis' approach will yield an equally impressive improvement when applied to our contrast enhancement and noise filtering algorithms.



(a) (b)

Fig. 4. (a) Multiplicative noise, $U(0.7, 1.0)$. (b) Multiplicative noise removed.



(a) (b)

Fig. 5. (a) Image corrupted by additive and multiplicative noise. (b) Restored.

In conclusion, image processing algorithms presented in this paper, based on considerations of the local image statistics, have a structure which makes them naturally suitable for parallel processing. Since the latter approach offers great computational economy, real or near real-time processing can be achieved. Future research in this area is to extend the method to image restoration of motion blur and other degradations characterized by local correlations around pixels.

Acknowledgment

The author wishes to thank Drs. I. Jurkevich and A. F. Petty for many helpful discussions and encouragements.

References

[1] T. S. Huang, Ed., *Picture Processing and Digital Filtering (Topics in Applied Physics)*, vol. 6. New York: Springer-Verlag, 1975.

[2] H. C. Andrews and B. R. Hunt, *Digital Image Restoration*. Englewood Cliffs, NJ: Prentice-Hall, 1977.

[3] A. Rosenfeld and A. C. Kak, *Digital Picture Processing*. New York: Academic, Aug. 1976.

[4] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME, J. Basic Eng.*, ser. D, vol. 82, pp. 35–45, 1960.

[5] A. K. Jain, "A semicausal model for recursive filtering of two-dimensional images," *IEEE Trans. Comput.*, vol. C-26, Apr. 1977.

[6] N. E. Nahi and T. Assefi, "Bayesian recursive image estimation," *IEEE Trans. Comput.*, vol. C-12, pp. 734–738, July 1972.

[7] P. J. Ketcham, R. W. Lowe, and J. W. Weber, "Real time image enhancement techniques," in *Proc. Seminar on Image Processing*, Pacific Grove, CA, Feb. 1976.

[8] R. Wallis, "An approach to the space variant restoration and enhancement of images," in *Proc. Symp. on Current Mathematical Problems in Image Science*, Naval Postgraduate School, Monterey, CA, Nov. 1976.

[9] W. Frei, "Image enhancement by histogram hyperbolization," *Comput. Graphics Image Processing*, vol. 6, pp. 286–294, June 1977.

[10] A. E. Bryson and Y. C. Ho, *Applied Optimal Control*. Waltham, MA: Blaisdell, 1969.

[11] N. E. Nahi and M. Naraghi, "A general image estimation algorithm applicable to multiplicative and non-Gaussian noise," Univ. of Southern California, Image Processing Inst. Semi-Annual Tech. Rep. 620, Sept. 30, 1975.

TAB D

United States Patent [19]

Sabri

[11] Patent Number: 4,528,584

[45] Date of Patent: Jul. 9, 1985

[54] BILEVEL CODING OF COLOR VIDEO SIGNALS

[75] Inventor: Mohammed S. Sabri, Beaconsfield, Canada

[73] Assignee: Northern Telecom Limited, Montreal, Canada

[21] Appl. No.: 446,608

[22] Filed: Dec. 3, 1982

[51] Int. Cl.³ H04N 9/32

[52] U.S. Cl. 358/12; 358/13

[58] Field of Search 358/12, 13

[56] References Cited

U.S. PATENT DOCUMENTS

4,467,346 8/1984 Mori 358/13

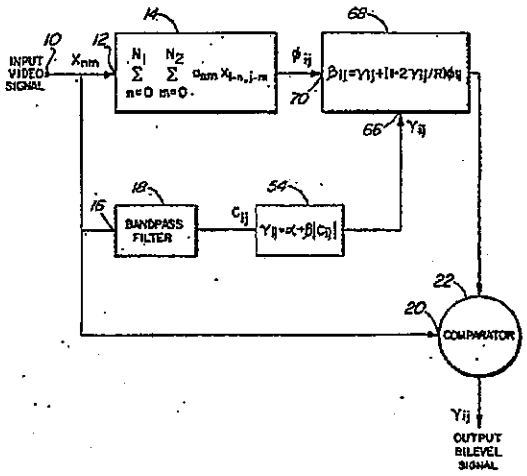
Primary Examiner—Richard Murray

Attorney, Agent, or Firm—Thomas Adams

[57] ABSTRACT

Bilevel coding of color video signals, for example video teleconference or NTSC broadcast television signals, is used to reduce the required storage capacity or transmission channel bandwidth. Each multi-level or continuous tone picture element in a frame is compared to a threshold and assigned one of two values depending upon whether or not it exceeds the threshold value. The threshold is produced by averaging the luminance components of neighboring picture elements and constraining the average by a signal derived from the chrominance of the original video signal. A significant feature is that the use of the chrominance signal in this way obviates the need, common to known monochrome systems, of adding noise (or dither signal) to the signal with concomitant picture degrading artifacts.

25 Claims, 2 Drawing Figures



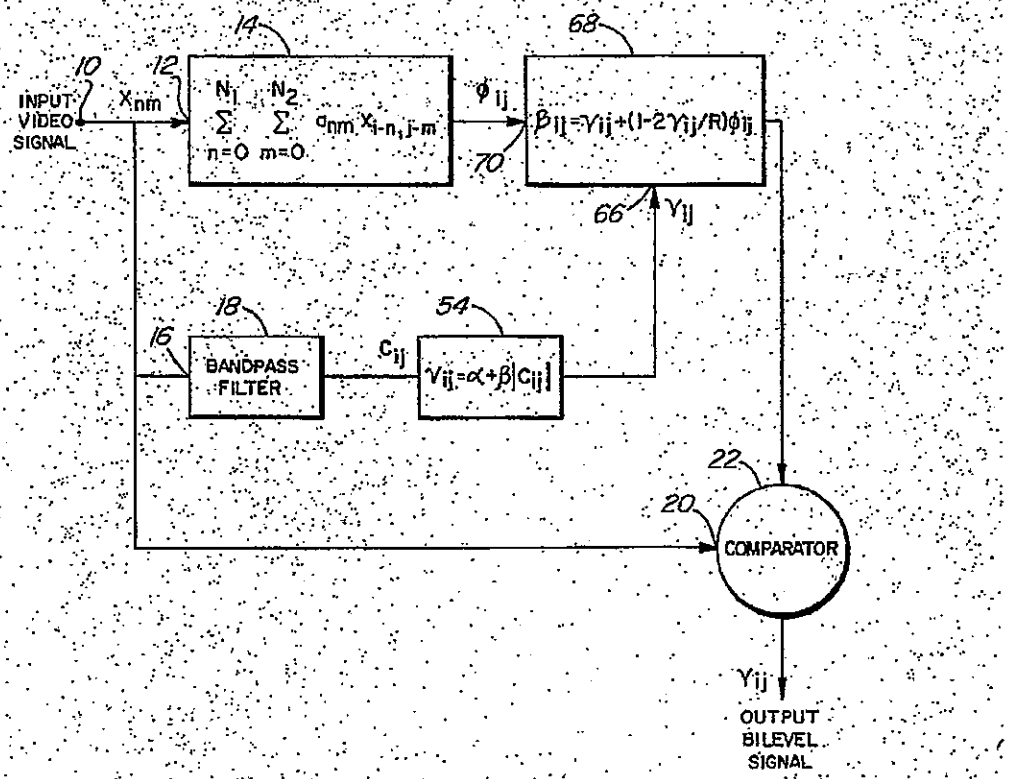

10
INPUT VIDEO SIGNAL
X_{nm}
12
14
$\sum_{n=0}^{N_1} \sum_{m=0}^{N_2} a_{nm} x_{i-n,j-m}$
ϕ_{ij}
68
$\beta_{ij}=\gamma_{ij}+(1-2\gamma_{ij}/R)\phi_{ij}$
70
66
γ_{ij}
16
18
BANDPASS FILTER
c_{ij}
54
$\gamma_{ij}=\alpha+\beta|c_{ij}|$
22
20
COMPARATOR
γ_{ij}
OUTPUT BILEVEL SIGNAL

FIG. 1

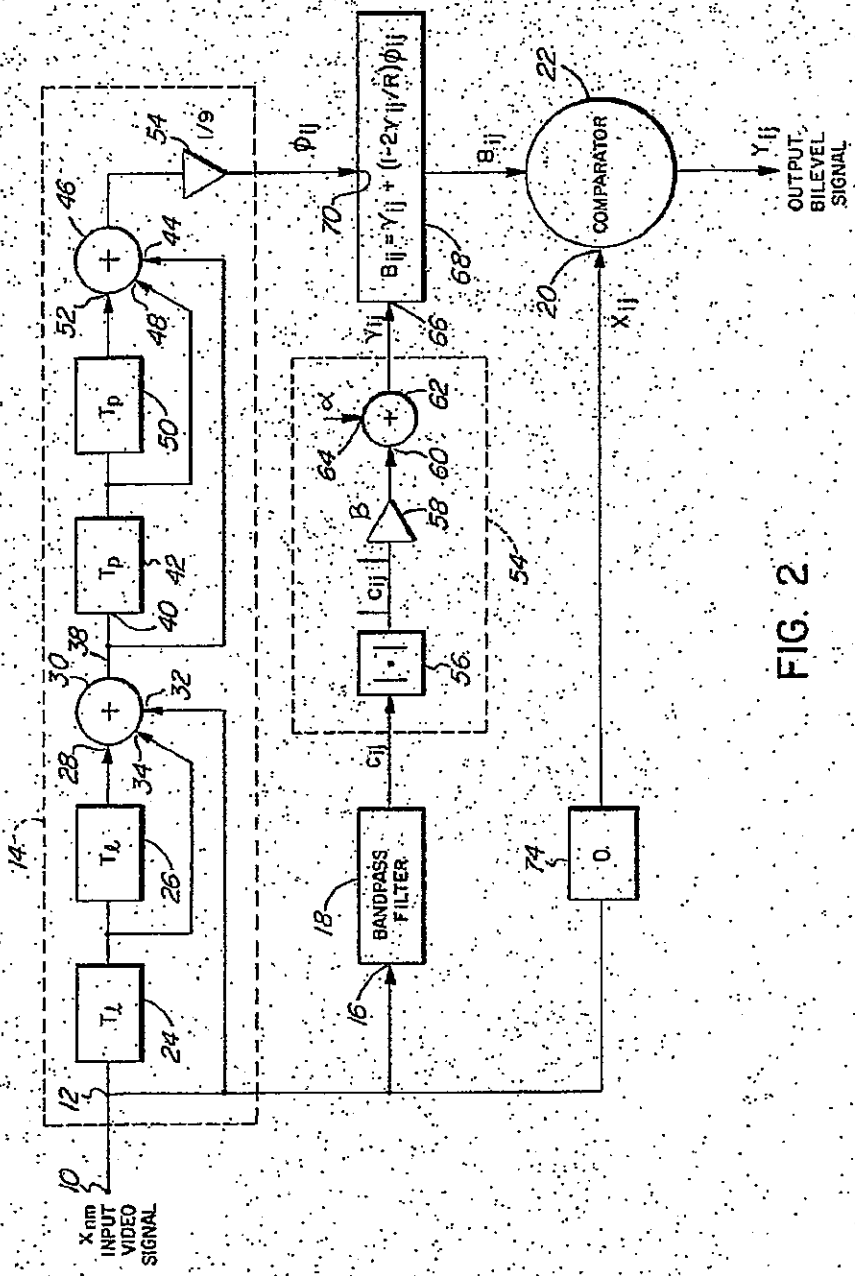

X_{nm} INPUT VIDEO SIGNAL
10
12
14
16
18
BANDPASS FILTER
20
22
COMPARATOR
OUTPUT BILEVEL SIGNAL
Y_{ij}
B_{ij}
ϕ_{ij}
X_{ij}
C_{ij}
T_p
T_l
1/9
α
β
0
$B_{ij}=Y_{ij}+(1-2Y_{ij}/R)\phi_{ij}$

FIG. 2

BILEVEL CODING OF COLOR VIDEO SIGNALS

The invention relates to video signal processing, particularly to a method and apparatus for bilevel coding or representation of colour video signals, and is especially applicable to composite television signals, for example NTSC broadcast colour television signals.

Whether a video signal originates as a broadcast television signal or otherwise, for example in an interactive visual communications system using video terminals and telephone lines, the need often arises to code the signal to reduce the required storage capacity or transmission channel bandwidth. Also in television programme production it is sometimes desirable to create a stylized representation or a sketch of the picture for special effects or animation. A type of process evolved for this purpose is so-called "bilevel coding (representation)". In such a process, each multi-level or continuous tone point in the frame, or picture element (Pel), is represented in bilevel form, i.e. it can assume only one of two states, typically high or low. Consequently only one digital bit is required to represent each picture element and the receiver can be very simple.

Basically bilevel coding involves comparing the picture element intensity with a threshold value and, depending upon whether or not it exceeds the threshold, assigning a corresponding output signal to either a high or low state. Generally the threshold value is chosen, and hence varied, according to the content of an area of the picture surrounding or adjacent to the picture element being coded at a particular instant. Determination of the threshold value affects the ability to preserve grey scale rendition, apparent contrast and detail in the picture.

Hitherto bilevel coding has generally been limited to monochrome signals. One known process, known as the "ordered dither" technique is disclosed in a paper entitled "Design of dither waveforms for quantized visual signals", by J. O. Limb. Bell System Technical Journal No. 48, 1969 pp. 2555–2582 and a paper entitled "An optimum method for two level rendition of continuous tone pictures", by B. E. Bayer, ICC conference record, 1973 pp. 26–11, 26–15.

In this process a set of predetermined position-dependent thresholds are used to derive the bilevel picture elements or a random dither signal, for example noise, is added to the multilevel picture. The process is primarily suited to bilevel displays since an objectionable flicker is usually perceived if the signal is displayed on a television monitor. Also, the process is not satisfactory for colour television signals due to excessive high frequency components in the chrominance bands which overlap with the chrominance signal and distort it.

An alternative process, known as the "constrained average technique" is disclosed in U.S. Pat. No. 3,961,134 issued June 1, 1976, which is hereby incorporated by reference. In this process the threshold value is chosen as a function of the local average within an area surrounding the point or picture element. To produce an apparent grey scale the process utilizes noise in the video signal. To alleviate the reliance upon an indeterminate parameter a controlled amount of noise (dither) is added to the video signal, in effect utilizing the ordered dither technique. Accordingly, the resulting picture suffers from the same problems of flicker and chrominance band distortion.

An object of the present invention is to eliminate or at least mitigate these problems in providing a process and apparatus for bilevel coding of a colour video signal.

According to one aspect of the present invention a process for bilevel coding of a colour video signal includes the steps of:

(i) deriving from a plurality of picture elements a first signal proportional to the luminance component of the colour video signal;

(ii) deriving a second signal proportional to the chrominance component of the colour video signal;

(iii) computing from the first and second signals a threshold value;

(iv) comparing a picture element value to the threshold value and deriving an output signal having one of two states in dependence upon whether or not the picture element value exceeds the threshold value.

In preferred embodiments of the invention the signal proportional to the luminance component is a weighted constrained average (ϕ_{ij}), computed, for example, in accordance with the equation

$$\phi_{ij} = \sum_{n=0}^{N_1} \sum_{m=0}^{N_2} a_{nm} X_{i-n,j-m}$$

where X is the pel value and a_{nm} is the weighting coefficient chosen to the signal proportional to the luminance.

Then the chrominance-proportional signal (C_{ij}), which acts as a dither signal producing an apparent grey scale, is used to derive a contrast enhancement factor X_{ij} thus:

$$\gamma_{ij} = \alpha + \beta |C_{ij}|$$

where α and β are constants. Conveniently α is in the range 5–20, preferably 10; and β is in the range 0.5–2, preferably 1.0. $|C_{ij}|$ denotes the absolute value or magnitude of C_{ij}.

The threshold value B_{ij} is then computed as:

$$B_{ij} = \gamma_{ij} + (1 - 2\gamma_{ij}/R)\phi_{ij}$$

Where γ_{ij} is the contrast enhancement factor and R is the maximum range of the video signal, for example 256 for an 8 bit digital signal.

According to another aspect, the invention comprises apparatus for implementing the process of the first aspect. Thus, said apparatus comprises:

means for deriving from a plurality of picture elements a first signal proportional to the luminance component of the colour video signal;

means for deriving a second signal proportional to the chrominance component of the colour video signal;

means for computing from the first and second signals a threshold value; and

means for comparing a picture element value to said threshold value to derive an output signal having one of two states in dependence upon whether the picture element value exceeds the threshold value or not.

Means for deriving the chrominance-proportional signal may conveniently comprise a bandpass filter having a pass band encompassing the chrominance subcarrier frequency f_{sc}. Such bandpass filter may be a one- or two-dimensional digital filter.

A particularly simple apparatus may be realized if the signal proportional to the luminance component is a weighted constrained ϕ_{ij} average of the form:

$$\phi_{ij} = \sum_{n=0}^{N_1} \sum_{m=0}^{N_2} a_{nm} X_{i-n,j-m}$$

and furthermore if the weighting coefficient a_{nm} is substantially unity and N_1 is an integer multiple of the subcarrier cycles. Thus if a sampling frequency of $3f_{sc}$ were used, N_1 would be multiples of three.

An exemplary embodiment of the invention will now be described with reference to the accompanying drawings, in which:

FIG. 1 is a schematic block representation of apparatus for bilevel coding of a colour video signal; and

FIG. 2 is a more detailed representation of the apparatus for a sampling frequency triple the colour subcarrier frequency.

It should be understood that although the input video signal can be analog or digital, for the specific embodiment the input signal is in digital form, for example 8 bits, and could be derived from a store, wherein it is stored still in digital form, or could be derived from a camera or other such source with an intervening analog/digital converter to convert it to the digital format required for input to the apparatus. The output bilevel signal will be decoded in the receiver. Since the receiver decoding circuit is known, it too is not shown in the drawing. Typically the receiver will comprise a standard monitor with a one bit digital/analog converter.

Referring to FIG. 1, apparatus for bilevel coding an NTSC composite colour video signal comprises an input terminal 10 to which is applied the digital video signal to be encoded sampled at three times the colour subcarrier frequency f_{sc}. The input terminal 10 is connected to, respectively, an input 12 of summing means 14, an input 16 of a bandpass filter 18 and an input 20 of a comparator 22. The summing means 14 serves to compute from the input picture elements X_{ij} a constrained weighted average ϕ_{ij} according to the formula:

$$\phi_{ij} = \sum_{n=0}^{N_1} \sum_{m=0}^{N_2} a_{nm} X_{i-n,j-m}$$

$X_{i-n,j-m}$ is the picture element preceding the particular element being encoded. As can be seen from FIG. 2, summing means 14 is adapted to sum nine preceding picture elements to produce the weighted constrained average ϕ_{ij}. Thus, summing means 14 comprises two delays 24 and 26, each equal to the duration of one line scan of the signal. The delays 24 and 26 are connected in series between the input 12 and one input 28 of an adder 30. The input 12 also is connected directly to an input 32 of the adder 30. A third input 34 of the adder 30 is connected to the output 36 of the first delay 24. The output of the adder 30, on line 38, is applied to the input 40 of a delay 42, equal to the duration of one picture element (Pel). Line 38 also is connected to an input 44 of a second adder 46. The output of the Pel delay 42 is applied to a second input 48 of the adder 46 and to a second Pel delay 50, the output of which is applied to a third input 52 of the adder 46. The output of adder 46 is connected to a divider 54, where the summed signal is divided by nine to give the signal ϕ_{ij} proportional to the luminance of the composite colour video signal X_{ij}. For this case the individual coefficients a_{nm} for each delay or Pel is equal to one ninth. Sampling at $3f_{sc}$, the sequence of values of each three successive picture elements will be Y+I,

$$Y - \frac{I}{2} + \frac{\sqrt{3}}{2} Q \text{ and } Y - \frac{I}{2} - \frac{\sqrt{3}}{2} Q.$$

where Y is the luminance component and I and Q are the chrominance components. It will be seen that when these are summed the resultant is proportional to luminance only.

As will be described more fully hereafter, other proportions can be used depending upon sampling frequency and number of samples.

Bandpass filter 18 is a one-or two-dimensional digital filter arranged to pass the chrominance signal, C_{ij}. The centre frequency of the passband is f_{sc}, where f_{sc} is the subcarrier frequency. The output of the filter 18 is connected to the input of means 54 for deriving a contrast enhancement factor proportional to the absolute value of C_{ij} plus a constant α. Means 54 comprises a device 56 for deriving the magnitude, (in effect a rectifier), the output of which is applied by way of an multiplier 58, having a multiplication factor β, to an input 60 of an adder 62. A second input 64 of the adder 62 is connected to a reference source having a value α. The output (γ_{ij}) of the adder 62 is connected to an input 66 of means 68 for computing the threshold value B_{ij} according to the equation:

$$B_{ij} = \gamma_{ij} + (1 - 2\gamma_{ij}/R)\phi_{ij}$$

Where γ_{ij} is the contrast enhancement factor and R is the maximum range of the video signal, for example 256 for an 8 bit digital signal.

A second input 70 of means 68 is connected to the output of divider 54. The output of computing means 68 is applied to a second input 72 of comparator 22. The input signal from terminal 12 is applied to the first input 20 of the comparator by way of a delay 74, equal to the delay of the bandpass filter 18. The output of the comparator 22 constitutes the output bilevel signal for storage and/or transmission.

As mentioned previously, the means 14 derives the weighted constrained average ϕ_{ij} from nine picture elements in a 3×3 matrix. The last element, in a conventional scan at the corner of the matrix, is the element X_{ij} applied also to the comparator 22 and hence compared with the threshold value B_{ij}. The output bilevel signal Y_{ij} will be high (S_1) if the element intensity X_{ij} exceeds B_{ij} and low (S_0) if it does not.

As mentioned previously, the values of a_{nm} can be chosen to render a signal proportional to luminance only, even though it has other than nine samples and a different sampling frequency. For example, at four times the colour subcarrier sampling frequency ($4f_{sc}$) the coefficients for a_{nm} can be:

$$\frac{1}{16}\begin{bmatrix} -1 & 0 & 2 & 0 & -1 \\ 2 & 0 & 12 & 0 & 2 \\ -1 & 0 & 2 & 0 & -1 \end{bmatrix}$$

It should be appreciated that although the specific description is for a digital video signal, the invention is not limited to digital signals but also comprehends analog signal processing. Then means for deriving the luminance-related signal could be a low pass filter, means for

deriving the contrast enhancement factor could be a rectifier, amplifier and adder or summing amplifier, and means for computing the threshold value B_{ij} would use suitable analog devices.

It will also be appreciated that for some applications, such as stylized single frame images, the invention need not be embodied using the precise circuit element described hereinbefore, but could be put into practice using a microprocessor.

With suitable modification the invention could be applied to component colour video signals, i.e. in which the luminance and chrominance components have previously been separated.

A significant advantage of using the chrominance signal to constrain the averaged luminance signal is that the video signal reproduced can be kept relatively free from objectionable artifacts such as flicker or distorted colour. It will be appreciated that to add noise as in the various dither techniques mentioned in the introduction will generally degrade picture quality.

What is claimed is:

1. Video signal processing apparatus comprising means for bilevel coding discrete picture elements of a colour video signal having a luminance component and a chrominance component, including:

means for deriving from a plurality of picture elements including at least one picture element other than the picture element to be encoded a first signal, (ϕ_{ij}), proportional to said luminance component;

means for deriving a second signal (C_{ij}) proportional to said chrominance component;

means for computing from said the first and second signals a threshold signal (B_{ij}) and

a comparator for comparing said picture element to be encoded with said threshold value and providing an output signal Y_{ij} having either one of two states in dependence upon whether or not said picture element exceeds said threshold value.

2. Video signal processing apparatus as defined in claim 1, wherein said means for deriving a luminance-proportional signal comprises means for computing a weighted average (ϕ_{ij}) of the luminance and chrominance components of a plurality of picture elements.

3. Video signal processing apparatus as defined in claim 2, wherein said means for computing is adapted to compute said weighted average ϕ_{ij} in accordance with the formula

$$\phi_{ij} = \sum_{n=0}^{N_1} \sum_{m=0}^{N_2} a_{nm} X_{i-n,j-m}$$

where a_{nm} is a weighting factor and $X_{i-n,j-m}$ adjacent picture elements' values.

4. Video signal processing apparatus as defined in claim 2, wherein said means for deriving said luminance-proportional signal is arranged to compute said signal from a plurality of neighbouring picture elements.

5. Video signal processing apparatus as defined in claim 4, wherein said plurality of neighbouring picture elements comprise a 3×3 matrix including said picture element to be encoded.

6. Video signal processing apparatus as defined in claim 5, wherein said means for deriving said luminance-proportional signal comprises two line delays connected serially between the input and an adder, the input being connected directly to one adder input, the output of the first delay connected to a second adder input and the second line delay connected to a third adder input.

7. Video signal processing apparatus as defined in claim 6, further comprising two picture element delays connected serially between the output of said adder and a first input of a second adder, the output of the first adder being connected to a second input of the second adder and the output of the first picture element delay being connected to a third input of the second adder.

8. Video signal processing apparatus as defined in claim 1, wherein said means for deriving a chrominance-proportional signal comprises a bandpass filter having a pass band to pass the chrominance signal.

9. Video signal processing apparatus as defined in claim 8, wherein said filter comprises a one- or two-dimensional digital filter.

10. Video signal processing apparatus as defined in claim 8 wherein said means for deriving a chrominance-proportional signal further comprises means responsive to the output of the filter for providing a contrast enhancement factor γ_{ij}.

11. Video signal processing apparatus as defined in claim 10, wherein said means for providing said contrast enhancement factor is arranged to do so in accordance with the formula $\gamma_{ij}=\alpha+\beta|C_{ij}|$ where $|C_{ij}|$ is the magnitude of C_{ij} and α and β are constants.

12. Video signal processing apparatus as defined in claim 11, wherein α is in the range 5 to 20.

13. Video signal processing apparatus as defined in claim 12, wherein α is equal to 10.

14. Video signal processing apparatus as defined in claim 11, wherein β is in the range 0.5 to 2.0.

15. Video signal processing apparatus as defined in claim 14, wherein β is equal to 1.

16. Video signal processing apparatus as defined in claim 1, wherein said means for computing said threshold signal B_{ij} is arranged to do so in accordance with the formula $B_{ij}=\gamma_{ij}+(1-2\ \gamma_{ij}/R)\phi_{ij}$ where γ_{ij} is a contrast enhancement factor related to said chrominance-proportional signal and R is the maximum range of the video signal.

17. A process for bilevel coding of a colour video signal having a luminance component and a chrominance component comprising the steps of:

(i) deriving from a plurality of picture elements including one or more picture elements other than the picture element to be encoded a first signal (ϕ_{ij}) proportional to said luminance component of the video signal;

(ii) deriving a second signal (C_{ij}) proportional to said chrominance component;

(iii) computing from said first and second signals (ϕ_{ij}) and (C_{ij}) a threshold value;

(iv) comparing said picture element to be encoded with said threshold value and providing an output having either one of two states in dependence upon whether or not said picture element exceeds said threshold value.

18. A process as defined in claim 17, wherein the step of deriving a luminance-proportional signal includes computing a weighted average of a plurality of picture elements.

19. A process as defined in claim 18, wherein said weighted average ϕ_{ij} is computed in accordance with the relationship

$$\phi_{ij} = \sum_{n=0}^{N1} \sum_{m=0}^{N2} a_{nm} X_{i-n,j-m}$$

where a_{nm} is a weighting coefficient and $X_{i-n,j-m}$ the values of said plurality of picture elements.

20. A process as defined in claim 19, wherein said plurality of picture elements comprises with the elements to be encoded a 3 by 3 matrix.

21. A process as defined in claim 19, wherein said threshold value B_{ij} is computed in accordance with the relationship $B_{ij} = \gamma_{ij} + (1-2\gamma_{ij}/R)\ \phi_{ij}$ where R is the maximum range of the video signal.

22. A process as defined in claim 17, wherein said signal C_{ij} is derived by extracting the chrominance signal by means of a bandpass filter whose centre frequency is at f_{sc}.

23. A process as defined in claim 22, wherein a contrast enhancement factor γ_{ij} is computed from the signal C_{ij} in accordance with the relationshp $\gamma_{ij} = \alpha + \beta|C_{ij}|$ where α is in the range 5 to 10, β is in the range 0.5 to 2.0, and $|C_{ij}|$ is the absolute value of C_{ij}.

24. A process as defined in claim 23, wherein α is equal to 10.

25. A process as defined in claim 22, wherein β is equal to unity.

* * * * *

TAB E

Feature enhancement of film mammograms using fixed and adaptive neighborhoods

Richard Gordon and Rangaraj M. Rangayyan

Digital techniques are presented for xerographylike enhancement of features in film mammograms. The mammographic image is first digitized using a procedure for gray scale dynamic range expansion. A pixel operator is then applied to the image, which performs contrast enhancement according to a specified function. The final transformation leads to either a positive or negative mode display as desired. We also present an adaptive neighborhood feature enhancement technique that enhances visibility of objects and details in an image. The availability of the enhanced images should aid diagnosis of breast cancer without requiring additional x-ray dose such as for xeromammography.

I. Introduction

While arguments regarding the relative merits and demerits of film-screen mammography and xeromammography continue,[1–4] it is generally conceded that the availability of both would greatly improve accuracy of diagnosis. However, xeromammography, while depicting clinical details more clearly with better contrast, also requires a much heavier radiation dose than modern film-screen methods (typically five times). A further disadvantage is its lower spatial resolution than film. We present here digital techniques for xerographylike contrast and feature enhancement of film mammograms, which should aid diagnosis without requiring the additional x-ray dose for xeromammography.

Our image processing operation begins with acquisition of a digital image of the mammogram covering the wide range of gray levels present. Since video systems used for image acquisition and digitization have a limited dynamic range, we use two images acquired at different illumination levels to generate a composite digital image, see Ref. 5 for details. A pixel operator is applied to the image which performs contrast enhancement according to a specified function. While implementation of the xerographic enhancement process is possible (see Refs. 6–9 for a description of the modulation transfer function of xerography; see also Refs. 10 and 11 for other techniques for enhancement of mammograms), our method has the flexibility of being able to perform any specified contrast enhancement operation. This feature enables experimentation with a variety of enhancement functions. The final step is transformation of the pixel values for display either in the positive or negative mode.

Contrast enhancement methods depend on differences between neighboring pixels. However, the values of these differences clearly depend on the magnification used. Thus the neighborhood size should be adjusted to the physical resolution. Contrast enhancement also depends on the sharpness of the gradient of density, which depends on the size of the objects and details present in the image. The same object digitized on a coarse raster would have a larger gradient than when digitized on a fine raster. Thus if an image with a wide range of gradients is to be enhanced, we have to use either a varying threshold for enhancement or change the neighborhood size. The former is undesirable, because noise will cause many spurious enhancement operations. The use of a larger neighborhood permits averaging over a number of pixels before differences are calculated, reducing spurious enhancements. We have therefore experimented with methods for adapting the neighborhood size to the local details and features on a pixel-by-pixel basis. The resulting images display features better than those obtained by fixed neighborhood operations. The method is applicable to all types of image and forms the basis for a new, general method for digital image processing.[12,13]

We believe our methods will aid better presentation of details available in conventional film mammograms. The availability of a xerographic representation of the same film mammogram with contrast enhancement (at no extra radiation) and the adaptive neighborhood image should be of considerable diagnostic value.

Both authors are with University of Manitoba, Winnipeg, Manitoba R3E 0W3; R. Gordon is in the Department of Radiology and R. M. Rangayyan is in the Department of Electrical Engineering.

Received 12 April 1983.
0003-6935/84/040560-05$02.00/0.

II. Methods

A. Image Acquisition

We use a Hamamatsu (Waltham, Mass.) C1000 TV camera with a Chalnicon tube to acquire video images of mammograms. The mammogram to be digitized is placed on an illuminator with intensity control. The video signal from the camera is digitized and stored in a Grinnell (San Jose, Calif.) GMR27 frame buffer. Our system has the capability of acquiring a 512 × 480 pixel image with 6-bit resolution (64 gray levels). The digitized image is next transferred to a Control Data Canada (Toronto) Cyber 171 computer via a Camac crate (Kinetic Systems, Lockport, Ill.). Image processing operations are performed by computer programs written in PASCAL (U. Minnesota) on the Cyber and the results transmitted back to the Grinnell system for display. We use a Conrac (Covina, Calif.) color TV monitor for displaying the images, and a Videoslide 35 camera unit (Lang Systems, Menlo Park, Calif.) to take photographs of the images.

As the video system has a limited gray level dynamic range, we acquire two digital images of the same mammogram at low and high illumination levels. The two images are then combined to generate a composite digital image having a wider gray level range; see Ref. 5 for details. By including a calibrated gray scale in the images and transforming pixel values to true optical densities, we also compensate for nonlinearities of the video system. Input images to the enhancement procedures may have pixel values in the optical density range of 0.05 (for white) to 3.04 (for black) as given by the calibrated gray scale used.

B. Contrast Enhancement

A pixel operator is next applied to the image for contrast enhancement. We define the local contrast between a pixel p and the average a of its eight neighbors in the 3 × 3 pixel matrix centered at p as

$$C = |p - a|/(p + a).$$

This definition gives the contrast measure C a range of 0–1. The contrast value is now enhanced according a specified function to a new value C'. A simple enhancement function is $C' = \sqrt{C}$, which increases lower values of contrast by a greater extent and preserves the range of 0–1 for C'. Our contrast enhancement procedure has the advantage over other standard neighborhood operations[14] that a straightforward implementation a any given enhancement function is possible. The new pixel value is then computed from C' and a as follows:

$$p' = a(1 + C')/(1 - C') \text{ if } p \geq a,$$

$$p' = a(1 - C')/(1 + C') \text{ if } p < a.$$

The final step is the transformation of the pixel values to the display range of 0–255. Here we have the choice of specifying either the positive or negative mode for display. If max and min are the maximum and minimum pixel values (optical density) in the image, we have the display pixel value p'' given by

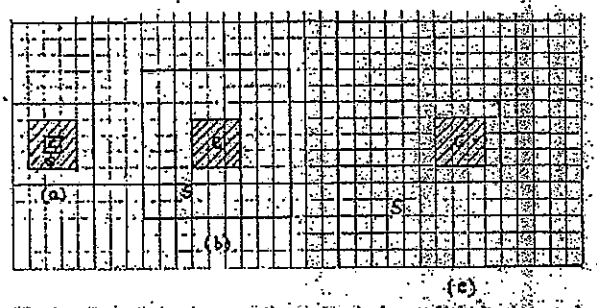


Fig. 1. For a dark square of size 3 × 3 pixels on a light background the contrast is zero in case (a) with center/surround of (1 × 1, 3 × 3); the contrast is maximum in (b) with center/surround of (3 × 3, 9 × 9); the contrast starts to fall with the next step of center/surround of (5 × 5, 15 × 15) as in (c). The center and surround regions are marked c and s in the figures, with the dark square hatched.

$$p'' = 255(p' - \min)/(\max - \min) \quad \text{positive mode,}$$

$$p'' = 255(\max - p')/(\max - \min) \quad \text{negative mode,}$$

with 0 for black and 255 for white in the display. Density windowing can also be incorporated at this stage for better visualization of selected density ranges.

C. Adaptive Neighborhood Processing

The standard neighborhood used in image processing (as above) has a center of one pixel and a surround of eight pixels,[14] if we make an analogy with visual receptive fields in vertebrates. We have found it desirable to retain this ratio of areas for adaptive neighborhood operations as well, and so consider neighborhoods with centers of size $m \times m$ and surrounds of size $3m \times 3m$, where m is any odd integer.[12] The center sizes are kept odd so that the center/surround regions can be centered about a pixel. Thus our neighborhood sizes are (1 × 1, 3 × 3), (3 × 3, 9 × 9), (5 × 5, 15 × 15), etc. The same definition for contrast as before is used, with p as the average of all pixels in the center region, and a as the average of all pixels in the surround region.

To determine the optimum neighborhood size to be used around a given pixel, we thought that a function giving contrast vs the size of the central region would be helpful. As an example, the contrast of a dark square on a light background will be zero when both the center and surround regions fall within the dark square. The contrast increases when the surround region takes in the background and peaks when the center region exactly covers the dark square. Further increase in size of the center region would reduce the contrast; see Fig. 1. We have thus used the first maximum of the contrast vs center size function as the indicator for optimum neighborhood size to be used at a given pixel location. The procedure for contrast enhancement is then as follows: At each pixel the dimensions of the center/surround regions that correspond to the first peak in the contrast are determined. The contrast value for the optimum size chosen is then increased by the square root function. The central pixel is then given a new value determined by the average intensity in the sur-

Fig. 2. Digitized mammogram.

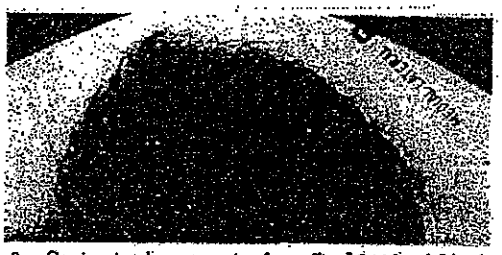


Fig. 3. Contrast enhancement using a fixed 3 × 3 neighborhood.

round region and the new contrast value as before. This is performed at all pixels in the image. The pixel values are then transformed to the display range.

III. Results and Discussion

Figure 2 shows a digitized film mammogram with expanded gray scale dynamic range (see Ref. 5 for details). (The spatial resolution, however, is poor due to digitization with an inadequate matrix of 461 × 269 points, which was the best possible with our equipment.) The corresponding contrast enhanced image using fixed neighborhoods is given in Fig. 3. As is evident, the enhanced image has better contrast and visibility of details. While the square root function has been used here for contrast enhancement, other functions could as well be tried depending on input image characteristics and desired enhancement.

Figure 4 shows a cluster of microcalcifications from a mammogram, enlarged and digitized using a macro-lens on the TV camera. (Dynamic range expansion was not required in this case due to the small area.) Application of the standard edge enhancement technique of adding the digital derivative (see Refs. 10 and 14) yielded Fig. 5, which shows marginal improvement. Figure 6 is the result of our contrast enhancement procedure with a fixed neighborhood size of 3 × 3 (the image is retained in the negative mode). Application of the new adaptive neighborhood feature enhancement procedure resulted in Fig. 7, which has better detail visibility than Figs. 4–6. Note that the uneven background in Fig. 7 is not noise but an enhancement of the weak background pattern in Fig. 4. As can be seen by comparing these figures, the adaptive neighborhood operation enhances objects and features as a whole, rather than enhancing edges only. A pictorial representation of the neighborhood size used is given in Fig. 8, and by comparing this range image with Fig. 4, we see

Fig. 4. Microcalcification cluster from a mammogram (enlarged and digitized).

Fig. 5. Edge enhancement by adding digital derivative, achieved by convolving with the 3 × 3 mask (−0.125, −0.125, −0.125; −0.125, 2, −0.125; −0.125, −0.125, −0.125), followed by clipping to the display range of 0–63.

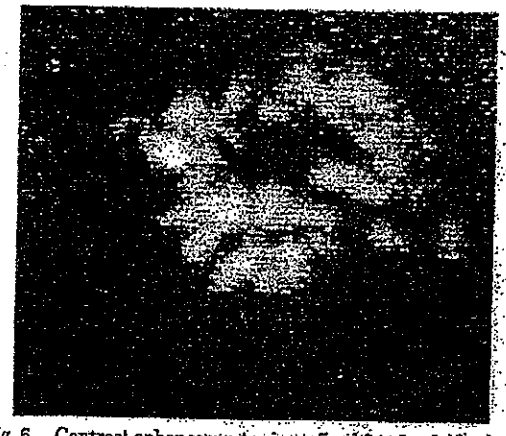


Fig. 6. Contrast enhancement using a fixed 3 × 3 neighborhood.

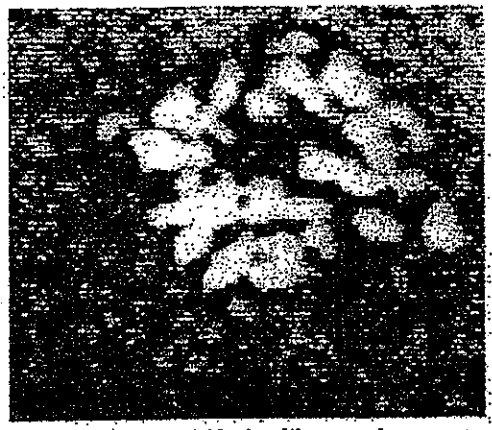


Fig. 7. Adaptive neighborhood feature enhancement.

Fig. 9. Adaptive neighborhood enhancement of mammogram in Fig. 2.

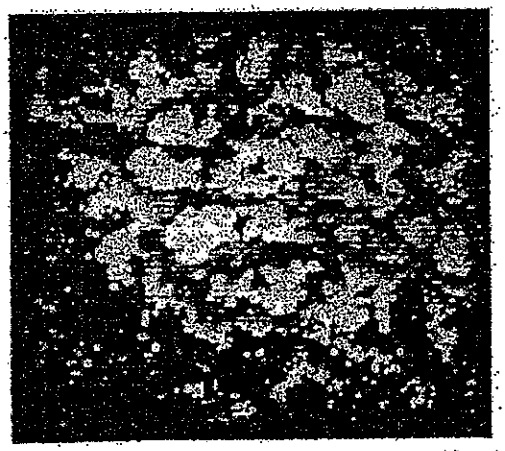


Fig. 8. Picture of neighborhood size used at different pixel locations. Whiter pixels in this range image represent pixels in the original image with larger adaptive neighborhoods.

Fig. 10. Range image for Fig. 2.

that the center size used at a pixel location correlates with the size of the object at that location. The images were limited in size to 150 × 150 pixels by the central memory available in our computer. Whole mammograms could be processed with larger computers or special purpose image processing systems.

While MTF (modulation transfer function) analysis of xerography and its simulations has been reported (e.g. Ref. 9), such a linear system theoretical analysis is, strictly speaking, not valid due to the nonlinear nature of the process (see also Ref. 11). Figures 9(a)–(c) give the Fourier transform magnitude spectra for the images in Figs. 4, 6, and 7. It is seen that the two enhancement procedures emphasize higher frequency components, corresponding to edge and contrast enhancement. Note that very small values toward higher frequencies are not seen in Fig. 9(a) due to transformation to the display range of 0–255 (integers). However, since all three spectra have been transformed to the same display range, they can be compared by visual analysis without much error. While we have not performed MTF analysis of the enhancement procedures, the high frequency emphasis nature of the processes is evident from the spectra shown.

Here we have presented new methods for contrast and feature enhancement which enhance the visibility of objects and details in an image. As with all contrast enhancement procedures, the gray scale range of the details is reduced. This may be viewed as an optimal mapping of the wide dynamic range of details present in the image to the limited gray scale range of the human visual system and of the display devices. The contrast enhancement function (the square-root function here) determines the dynamic range of the result: a less severe function could be used if a wider dynamic range is desired. We intend to replace the use of the first peak in the contrast function as an indicator of the optimum neighborhood size by a better scheme which will take into account noise in the function, as well as interactions between objects (ringing). We are now extending the same principles to other image processing techniques, generalizing the concept of pixel-independent image processing.[12]

Judging from our results, as well as those obtained by others, we believe strongly that xeromammography

should not be used at all. The increased dose required, for xeromammography by itself or as an addition to film mammography as commonly practiced, does not justify the enhanced features: digital image processing techniques applied to film mammograms, which have all details recorded due to the wider gray scale dynamic range, can readily provide similar results without additional x-ray dose. Enhancement techniques such as ours can also be viewed as redundancy reduction procedures from an information theoretical point of view.[13] A large scale clinical trial of the techniques with mammograms is contemplated.

Note added in proof: We have rewritten our program to use packed arrays enabling processing of larger images. The adaptive neighborhood enhancement of the mammogram in Fig. 2 is given in Fig. 9, with the corresponding range image in Fig. 10.

We thank John Dunning, Bill Paley, and the late Ronald Laramee of the Computer Department for Health Sciences, University of Manitoba Medical College, for providing image processing hardware and software support. We thank Douglas MacEwan, Department of Radiology, University of Manitoba, and Harold Standing and George Hardy of the Manitoba Mammography Unit, and David Harries, Chief Radiologist at Brandon General Hospital, for cooperation and assistance rendered. Thanks are due also to Aaron Stein and Curtis Quinn for developing software for display of images on the EMI Independent Viewing Station at the Health Sciences Center.

This work was supported by grants from the Manitoba Medical Services Foundation, the Natural Sciences and Engineering Research Council of Canada, and Control Data Corp. This paper was presented in part as a scientific exhibit at the Forty-Fifth Annual Meeting of the Canadian Association of Radiologists, Winnipeg, 29 May–3 June 1982. Richard Gordon also holds appointments in the Departments of Radiology, Electrical Engineering, and Zoology.

References

1. J. J. Pagani, L. W. Bassett, R. H. Gold, J. Benedetti, R. D. Arndt, J. Linsman, and R. L. Scanlan, Am. J. Roentgenol. 135, 141 (1980.
2. L. Stanton, T. Villafana, J. L. Day, D. A. Lightfoot, and R. E. Stanton, Radiology 132, 455 (1979).
3. L. N. Rothenburg, R. L. A. Kirch, and R. E. Snyder, Radiology 117, 701 (1975).
4. J. N. Wolfe, R. P. Dooley, and L. E. Harkins, Cancer 28, 1569 (1971).
5. R. M. Rangayyan and R. Gordon, "Expanding the Dynamic Range of X-Ray Videodensitometry for Digital Mammography and Teleradiology," Appl. Opt. 00, 000 (1983), submitted.
6. G. H. Zeman, G. U. V. Rao, and F. A. Osterman, Radiology 119, 689 (1976).
7. J. W. Boag, Phys. Med. Biol. 18, 3 (1973).
8. H. E. J. Neugebauer, Appl. Opt. 4, 453 (1965).
9. R. A. Kilgore, E. C. Gregg, and P. S. Rao, Opt. Eng. 13, 180 (1974).
10. M. B. McSweeney, P. Sprawls, and R. L. Egan, Am. J. Roentgenol. 140, 9 (1983).
11. H. J. Trussell, Proc. IEEE 69, 615 (1981).
12. R. Gordon and R. M. Rangayyan, "Pixel-Independent Image Processing," in preparation.
13. R. Gordon and R. M. Rangayyan, "Radiographic Feature Enhancement, Information Content and Dose Reduction in Mammography and Cardiac Angiography," in *Proceedings, Fifth Annual Conference on Frontiers of Engineering and Computing in Health Care* (IEEE-EMBS), Columbus, Sept. 1983, pp. 161–165.
14. A. Rosenfeld and A. C. Kak, *Digital Picture Processing* (Academic, New York, 1982).

TAB F

United States Patent [19]

Chen et al.

[11] **Patent Number: 4,789,933**

[45] **Date of Patent: Dec. 6, 1988**

[54] **FRACTAL MODEL BASED IMAGE PROCESSING**

[75] Inventors: **Victor C. Chen**, Richmond Hts., Ohio; **Mike M. Tesic**, Los Altos, Calif.

[73] Assignee: **Picker International, Inc.**, Highland Hts., Ohio

[21] Appl. No.: **19,568**

[22] Filed: **Feb. 27, 1987**

[51] **Int. Cl.**[4] **G06F 15/42**

[52] **U.S. Cl.** **364/413.13**; 382/47; 382/54

[58] **Field of Search** 382/27, 47, 54; 358/166, 167; 364/414

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,386,528	6/1983	Engle	358/112 X
4,633,503	12/1986	Hinman	382/47
4,665,551	5/1987	Sternberg	382/27 X
4,694,407	9/1987	Ogden	364/728 X
4,703,353	10/1987	David	382/47 X

OTHER PUBLICATIONS

"Analysis and Interpolation of Angiographic Images by Use of Fractals" by T. Lundahl, et al., IEEE, 1985, pp. 355–358.

"Digital Image Enchancement: A Survey" by Wang, et al. Computer Vision, Graphics, and Image Processing, 24, 363–381 (1983).

"Digital Image Processing by Use of Local Statistics" by Jong-Sen Lee, Naval Research Laboratory, pp. 55–61.

"Comparison of Interpolating Methods for Image Resampling" by Parker, et al., IEEE Transactions on Medical Imaging, vol. MI-2, No. 1, Mar. 1983, pp. 31–39.

"Nonstationary Statistical Image Models (and Their Application to Image Data Compression" by Hunt, 1980.

"Fractal Based Description of Natural Scenes" by Alex Pentland, IEEE 1983, pp. 201–209.

"A Note on Using the Fractal Dimension for Segmentation" by Medioni, et al, IEEE 1984, pp. 25–30.

"Computer Rendering of Stochastic Models" by Fournier, et al. Communications of the ACM, Jun. 1982, vol. 25, No. 6, pp. 371–384.

Primary Examiner—Jerry Smith
Assistant Examiner—Steven G. Kibby
Attorney, Agent, or Firm—Fay, Sharpe, Beall Fagan, Minnich & McKee

[57] **ABSTRACT**

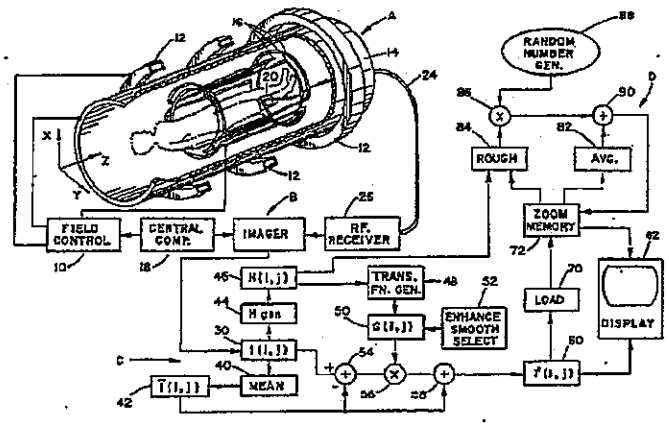
A medical diagnostic apparatus (A) generates medical diagnostic data which is reconstructed by an imager (B) into an electronic image representation. The electronic image representation includes an array of digital pixel values which represent a gray scale intensity of a man-readable image displayed on a video monitor (62). An image improving circuit (C) replaces each pixel value I(i,j) with an improved pixel value I'(i,j) defined as follows:

$$I'(i,j)=G(i,j)[I(i,j)-\bar{I}(i,j)]+\bar{I}(i,j),$$

where G(i,j) is a transfer function uniquely defined for each pixel and $\bar{I}$ is the mean of pixel values of neighboring pixels. The transfer function is based on a self-similarity value which is derived by comparing (i) a variation between the pixel value I(i,j) and pixel values in a first surrounding ring with (ii) a variation between the pixel value I(i,j) and pixel values in a second, larger surrounding ring. A zoom circuit (D) enlarges a selected portion of an improved image. Empty or unfilled pixel values I(inter) are interpolated from a combination of an average of neighboring pixel values I(avg) and a fractal value. The fractal value is a combination of a random number and a weighting factor R determined in accordance with the average self-similarity value of the neighboring filled pixels, i.e.:

$$I(\text{inter})=I(\text{avg})+R\cdot(\text{random\#}).$$

20 Claims, 2 Drawing Sheets



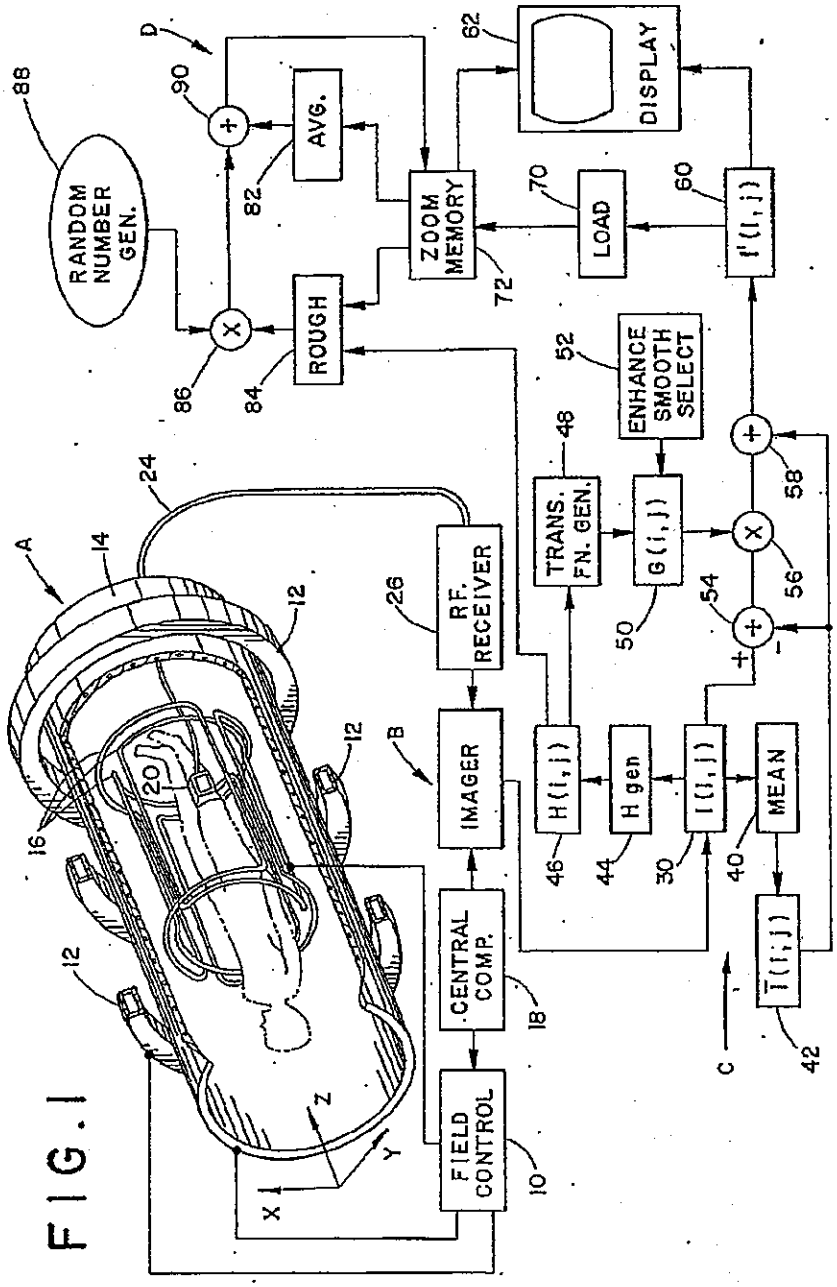

DISPLAY
62
D
90
AVG.
82
88
RANDOM NUMBER GEN.
ZOOM MEMORY
72
LOAD
70
60
I'(i,j)
ROUGH
86
84
ENHANCE SMOOTH SELECT
52
48
TRANS. FN. GEN.
G(i,j)
58
56
54
50
24
A
14
12
26
RF. RECEIVER
B
IMAGER
H(i,j)
H gen
I(i,j)
MEAN
46
44
30
40
20
16
CENTRAL COMP.
18
Ī(i,j)
42
C
FIELD CONTROL
10
Z
Y
X

FIG. 1

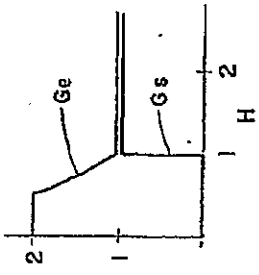


FIG. 3

FIG. 4

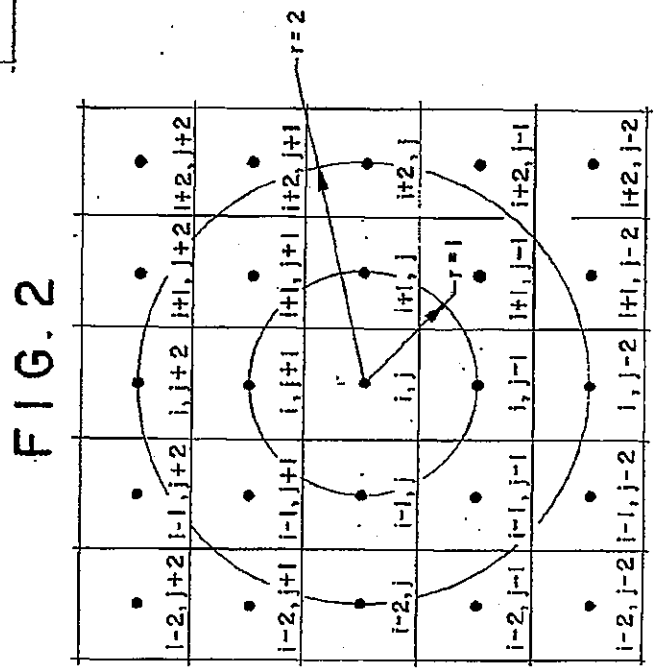


FIG. 2

FRACTAL MODEL BASED IMAGE PROCESSING

BACKGROUND OF THE INVENTION

The present invention relates to the art of image processing. It finds particular application in conjunction with image enhancement, image smoothing, image zooming and other image improvement techniques for magnetic resonance images and will be described with particular reference thereto. It is to be appreciated, however, that the present invention is also applicable to enhancing, improving and enlarging digital x-ray images, computed tomographic images, nuclear camera images, positron emission scanners, and the like.

Medical diagnostic images have commonly been subject to image degradation from noise, system imperfections, and the like. Various image processing techniques have been utilized to remove the effects of the noise and to highlight some specified features. See for example "Digital Image Enhancement: A Survey" Wang, et al., Computer Vision, Graphics, and Image Processing, Vol. 24, pages 363–381 (1983). In one technique, each pixel was adjusted in accordance with the mean of surrounding pixels and the variation or difference between the pixel value and the local mean (the average of the surrounding pixels). The enhanced pixel value $g'(i,j)$ was a weighted average of the local mean and the variation:

$$g'(i,j)=\bar{g}(i,j)+k[g(i,j)-\bar{g}(i,j)] \quad (1),$$

where $\bar{g}(i,j)$ was the local mean, $g(i,j)-\bar{g}(i,j)$ was the variation, and k was a constant that weighted the relative contributions therebetween. It is to be appreciated that when k was set larger than 1, the variation, hence the fine details were magnified. As k was set smaller, the image was smoothed or blurred as if acted upon by a low-pass filter. At the extreme at which k was set equal to zero, each pixel value was replaced by the local mean of the neighboring pixel values.

One of the drawbacks in this technique resided in selecting an appropriate value for the weighting factor k. The smaller k was set, the more the image was blurred and the more difficult it became to extract accurate diagnostic information. As k was set larger, edges and fine details, including noise, became enhanced. Frequently, in a medical image, the selected weighting factor k was too large for some regions and too small for other regions.

"Digital Image Enhancement and Noise Filtering by Use of Local Statistics" by J. S. Lee, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 2, pages 165–168 (1980), recognized that different weighting factor k could be selected for each pixel to be enhanced. Specifically, Lee suggested setting the k for each pixel equal to the square root of the ratio of a preselected desirable local variance to the actual local variance of the selected pixel. Although Lee's weighting factor achieved better resultant images than the constant weighting factor, there was still room for improvement.

Another problem with medical diagnostic images resided in the blurring of enlarged or zoomed images. Typically, the diagnostic image included a fixed number of pixel values, e.g. a 256×256 pixel value matrix or array. When the image or a pattern thereof was enlarged each pixel could be displayed as a larger rectangle or additional intervening pixel values must be generated. For example, when a 256×256 array was enlarged to a 512×512 array, no data existed for alternate lines and for alternate columns of the 512×512 matrix. Commonly, the missing matrix values were interpolated by linear averaging the nearest neighboring pixel values. However, averaging adjacent pixels tended to blur the resultant enlarged image.

Another interpolation technique for zooming is described in "Analysis and Interpretation of Angiographic Images by Use of Fractals", T. Lundahl, et al., IEEE Computer in Cardiology, page 355–358 (1985). In Lundahl's technique, a global or image wide fractal model was utilized to derive the missing, intervening pixel values of an enlarged digital angiographic image. Lundahl, et al. first calculated a global fractal dimension for the entire image which described the average roughness or smoothness of the intensity surface of the entire image. The interpolated pixel values were based on the average of neighboring pixel values plus or minus a function of global fractal dimension. The plus or minus sign was chosen at random. One of the drawbacks in their technique is that the interpolated value was always different from the neighboring average by a function of the global fractal dimension when in fact, in the real world there is a certain chance for the interpolated value to be equal to the average of the neighboring values. While the addition or subtraction of a function of the global fractal dimension value made the image more realistic and easier to view, the interpolated pixel values were inaccurate and could cause an erroneous diagnosis. Particularly in low pixel value regions of an image with a large global fractal dimension, fictious image details could be generated during enlargement. This was caused by their use of the global fractal dimension in doing local interpolation. This is the other drawback of their technique.

The present invention is based on a new image fractal model, which describes images much more like real-world images. By the use of the fractal model, the processed medical image looks much more natural than the one by the use of traditional techniques.

The fractal model is based on the theory of fractal brownian motions developed by B. Mandelbrot. This theory provided a useful model for description of real-world surface. See 'Fractal: Form, Chance, and Dimension' and 'The Fractal Geometry of Nature', B. Mandrelbrot, W. H. Freeman and Company (1977) and (1982) respectively.

SUMMARY OF THE INVENTION

In accordance with one aspect of the present invention, a method of diagnostic imaging is provided. An electronic image representation which includes an array of pixel values is generated. A self-similarity value is generated corresponding to each pixel value. Each self-similarity value varies in accordance with the variations between the corresponding pixel value and the surrounding contiguous pixel values. Each pixel value of the image representation is replaced by a weighted combination of the pixel value to be replaced and the average of the surrounding pixel values. The combination is weighted in accordance with the corresponding self-similarlity value such that for enhancement the more similar the pixel value is to its surrounding contiguous pixel values, the more heavily the pixel value is weighted, i.e. the less enhancement is taken, and the less similar the pixel value is to its surrounding, contiguous

pixel values, the more heavily the variation value is weighted. However, for smoothing, the more similar the pixel value is to its surrounding contiguous pixel values, the more heavily the average of the surrounding pixel values is weighted and the less similar the pixel value is to its surrounding pixel values, the more heavily the pixel value is weighted.

More specific to the preferred embodiment, the self-similarity value is selected in accordance with a ratio of (i) a first averaging difference between the given pixel value and the pixel values in a first surrounding ring and (ii) a second averaging difference between the given pixel value and the pixel values in a second surrounding ring. In this manner, the self-similarity factor varies in accordance with the ratio of the averaging difference between the given pixel value and two concentric rings therearound.

In accordance with another aspect of the present invention, a method of diagnostic imaging is provided. An electronic image representation is defined by an array of pixel values, a fraction of the pixel values are generated from medical diagnostic data and the remaining pixel values are determined by interpolating the diagnostic data based pixel values. For each of a plurality of regions of the image, a roughness factor which varies with the degree of variation among pixel values within the region is determined. The roughness is largest in areas with large variations and smallest in homogeneous regions. Each interpolated pixel value is set equal to an average of surrounding diagnostic data based pixel values plus the product of the roughness factor and a Gaussian random number. In this manner, the interpolated pixel values are substantially equal to the average of neighboring diagnostic data based pixel values in substantially homogeneous regions. In regions with widely varying pixel values, the interpolated pixel values may vary more significantly from the neighboring value average.

In accordance with another aspect of the present invention, an apparatus is provided for processing diagnostic images. An image data memory means stores a plurality of pixel values based on diagnostic data. A self-similarity value calculating means calculates a self-similarity value corresponding to each pixel value stored in the memory means. The self-similarity values are determined in accordance with a ratio of the differences between the corresponding pixel value and the pixel values in at least two surrounding rings of pixel values. A mean value determining means determines for each pixel value a corresponding mean value based on the average of neighboring pixel values. A filtering means combines each pixel value with the corresponding mean value with the relative weight of the pixel value and the corresponding mean value being weighted in accordance with the corresponding self-similarity value.

In accordance with another aspect of the present invention, an apparatus for processing diagnostic images is provided. A diagnostic imaging means generates a plurality of diagnostic data based pixel values which are stored at a predetermined fraction of the pixel of an image memory. The unfilled image memory pixels are denoted as interpolated pixels. An averaging means averages diagnostic data based pixel values neighboring each interpolated pixel. A roughness factor determining means determines a roughness factor in the neighborhood of each interpolated pixel, which roughness factor varies in accordance with the variation of diagnostic data based pixel values in the region. A random number generator generates Gaussian random numbers. A combining means combines the random number generated by the random number generator with the roughness factor to produce a fractal value and combines the fractal value with the average from the averaging means to generate an interpolated pixel value. An interpolating means returns each interpolated pixel value to the image memory for storage at the corresponding interpolated pixel.

One advantage of the present invention is that it generates medical diagnostic images of improved viewability.

Another advantage of the present invention is that it provides an improved pixel selective filtering function for automatically smoothing noise an enhancing resultant images.

Another advantage of the present invention is that it enlarges images while retaining the same apparent sharpness to the viewer's eye.

Still further advantages of the present invention will become apparent to those of ordinary skill in the art upon reading and understanding the following detailed description of the preferred embodiment.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention may take form in various steps and arrangements of steps and in various components and arrangements of components. The drawings are for purposes of illustrating a preferred embodiment and are not to be construed as limiting the invention.

FIG. 1 is a diagrammatic illustration of a medical diagnostic imaging system in accordance with the present invention;

FIG. 2 diagrammatically illustrates calculation of a self-similarity factor in a 5×5 pixel region about a pixel (i,j);

FIG. 3 is a diagrammatic illustration of a preferred transfer function G(i,j); and,

FIG. 4 is a diagrammatic illustration of an interpolation technique around the pixels of the zoom memory.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

With reference to FIG. 1, a medical diagnostic apparatus A generates medical diagnostic data which is reconstructed by an imager B into an electronic image representation. An image filter and enhancement circuit C operates on the electronic image representation to improve the image quality and viewability thereof. A zoom circuit D selectively enlarges the resultant image representation or regions thereof.

Although a magnetic resonance imager is illustrated, the medical diagnostic apparatus A may be a computerized tomographic scanner, a digital x-ray apparatus, a positron emission scanner, a nuclear camera, or other diagnostic apparatus which generates data that is able to be reconstructed into an image representative of a region of an examined patient or subject. The illustrated magnetic resonance imager includes a field control means 10 which controls a main, homogeneous polarizing magnetic field through an image region generated by electromagnets 12. The field control means 10 also controls gradient magnetic fields created across the image region by gradient field coils 14 to provide spatial encoding, phase encoding, and slice select gradients. The field control means 10 further generates radio frequency electromagnetic excitation signals which are

applied to excitation coils 16 to excite resonance of dipoles in the image region. A central computer 18 controls the relative timing and strengths of the gradient and radio frequency electromagnetic fields.

A receiving coil 20 is surface mounted on the patient to receive magnetic resonance signals generated by resonating dipoles in the image region. A flexible cable 24 conveys the magnetic resonance signals to radio frequency receiver 26. The radio frequency receiver 26 supplies magnetic resonance medical diagnostic data to the imager B.

The imager B under control of the central computer 18 reconstructs the medical diagnostic data into an electronic image representation. More specifically, the imager B reconstructs digital pixel values, each pixel value corresponding to a preselected subregion or pixel of the image region. The algorithm implemented by the imager is selected in accordance with the medical diagnostic apparatus selected. Conventionally, the pixels or subregions are arranged in a rectangular array (i,j) with the corresponding pixel values I(i,j) being arranged in a like rectangular array. The electronic image representation, particularly the rectangular array of pixel values I(i,j), is stored in a first image memory 30.

The image filtering and enhancing circuit C includes a pixel value averaging means 40 which generates a matrix of averaged or mean pixel values for storage in a mean value memory 42. Each mean pixel value I(i,j) is set equal to the average of the pixel values in the neighborhood of a corresponding pixel value of the first image memory 30. Preferably, the pixel mean value means 40 determines each pixel mean value as follows:

$$I(i,j) = \frac{1'}{(2l+1)^2} \sum_{x=0}^{l} \sum_{y=0}^{l} I(i \pm x, j \pm y), \tag{2}$$

where l is indicative of the size of the neighborhood over which the mean value is averaged. For example, as illustrated in FIG. 2, for l=2, a 5×5 neighborhood is defined in which the mean value represents the average of the 25 pixel values in the neighborhood. Optionally, other averaging algorithms may be utilized. For example, a 3×3 neighborhood may be selected. As yet another option, the average may include the average of the surrounding pixel values in the neighborhood but not the pixel value itself, e.g. the 24 surrounding pixel values of a 5×5 neighborhood. The surrounding pixel values may be weighted inversely with radial distance, or the like.

A self-similarity parameter generator 44 derives a self-similarity value H(i,j) corresponding to each pixel value for storage in a self-similarity value memory 46. In Euclidean geometry, th Euclidean dimension, E, is an integer. For example, E=2 for lines and E=3 for surfaces. In topological geometry, the topological dimension, Dt, is also an integer, but Dt=1 for lines and Dt=2 for surfaces. In fractal geometry, the fractal dimension, D, need not be an integer, although it is commonly a real number. The fractal dimension is equal to or greater than the topological dimension and less than corresponding Euclidian dimension. For example, $1 \leq D \leq 2$ for lines and $2 \leq D < 3$ for surfaces.

The fractal dimension may be viewed as a measurement of the degree of irregularity or roughness. As the fractal dimension becomes larger and approaches the Euclidean dimension, the surface becomes more rough; as the fractal dimension becomes smaller and approaches the topological dimension, the surface becomes smoother.

The self-similarity parameter H is equal to the difference in the Euclidean and fractal dimensions, i.e.

$$H = E - D \tag{3}$$

Thus, the smoothest surface is defined when H=1 and the surfaces become rougher as H becomes less than 1.

In the present invention, the self-similarity parameter for each pixel, H(i,j) is determined by monitoring the neighborhood or environment around the corresponding pixel. The greater the variation between a given pixel value and the surrounding pixel values, the smaller the self-similarity parameter. The more uniform the neighborhood, the closer the self-similarity parameter approaches one.

With reference to FIG. 2, the self-similarity parameter is derived using a convolution type function. Specifically, for a given pixel, (i,j), the averaging difference between the given pixel value I(i,j) and pixel values in a first surrounding ring is compared with the averaging difference between the given pixel value and pixel values in a second surrounding ring. For rings of radius m and n, the definition of a scaler Brownian function provides:

$$(n/m)^H = \frac{k(n;i,j)}{k(m;i,j)}, \tag{4}$$

$$k(n;i,j) = \left(\sum_{(n-1)<r\leq n} |I(i,j) - I(k,l)| \right) / N, \tag{5}$$

where

N=Number of pixels whose centers fall in the ring between n and n−1.

and

$$r^2 = (i-k)^2 + (j-l)^2 \tag{6}$$

An expression of H can be obtained by taking the base-10 logarithm of both sides of Equation (4):

$$H \log(n/m) = \log \left(\frac{k(n;i,j)}{k(m;i,j)} \right) \tag{7}$$

or

$$H(i,j) = \frac{\log k(n;i,j) - \log k(m;i,j)}{\log(n) - \log(m)} \tag{8}$$

Thus, a self-similarity parameter or value H is determined in accordance with a ratio of two differences. The two differences are (1) the difference between the logarithm of the function k at the nth ring and the logarithm of the function k at the mth ring and (2) the difference between the logarithm of n and the logarithm of m.

In the embodiment illustrated in FIG. 2, n=2 and m=1. For pixel (i,j), illustrated in FIG. 2, k(2;i,j) is the average of the differences between the pixel value I(i,j) and the pixel values corresponding to each of pixels (i,j+2), (i−1,j+1), (i−2,j), (i−1,j−1), (i,j−2), (i+1,j−1), (i+2,j), and (i+1,j+1). Similarly, with the radius of the inner ring set equal to one, k(1;i,j) includes the average of the differences between the intensity at

pixel (i,j) and the intensities corresponding to each of pixels (i,j+1), (i−1,j), (i,j−1), and (i+1,j). That is, k(1;i,j) is the average of the differences between the pixel intensity or pixel value of pixel (i,j) and each of the pixels whose centers line on or within the radius of one interpixel spacing, r=1, and k(2;i,j) includes the difference between the value of pixel (i,j) and the pixel value corresponding to each pixel whose center lies outside of the r=1 ring and on or inside the r=2 ring.

It is to be appreciated that although the two rings are illustrated in the preferred embodiment as being continuous, a gap may be defined between the two rings. For example, the first ring might include those pixels whose centers fall between ¾ and 1¼ of the interpixel spacing and the second ring might include those pixels whose centers fall in the ring between 1¾ and 2¼ times the interpixel spacing. As another example, the pixel value of each pixel whose rectangular area falls even partially within a given ring may be determined and the difference may be weighted in accordance with the percentage of the pixel area which falls within the given ring. Optionally, additional rings may be defined concentrically around the first and second rings. Preferably, the effect of additional outer rings is reduced inversely with the larger radius of the outer rings. In this manner, the self-similarity value for each pixel is derived from a convolution type function.

With continuing reference to FIG. 1 and further reference to FIG. 3, a transfer function means 48 derives a transfer function for each pixel G(i,j) which may be stored in a transfer function memory means 50. In the preferred embodiment, the transfer function may either (i) enhance the image, i.e. accentuate edge effects or (ii) smooth the image, i.e. reduce noise effects for a more uniform image. For enhancement, an enhancement transfer function $G_e(i,j)$ is utilized. In the preferred embodiment, the enhancement transfer function is selected in accordance with:

$$G_e(i,j) = \begin{cases} 2 & \text{for } H(i,j) < 0.5 \\ 1/H(i,j) & \text{for } 0.5 \leqq h(i,j) \leqq 1.0 \\ 1 & \text{for } H(i,j) > 1.0 \end{cases} \quad (9)$$

For smoothing, a smoothing transfer function $G_s(i,j)$ is selected for each pixel in accordance with:

$$G_s(i,j) = \begin{cases} 1 & \text{for } H(i,j) > 1.0 \\ 0 & \text{for } H(i,j) \leqq 1.0 \end{cases} \quad (10)$$

An operator control 52 enables the operator to select between the described enhancement and smoothing transfer functions or other preselected transfer functions.

It is to be appreciated that the image memory 30, the mean value memory 42, the self-similarity value memory 46, and the transfer function memory 50 are for purposes of illustration. The values described as stored in these memories may be stored as described, calculated in real time, or a combination of both.

In the preferred embodiment, the image smoothing-/enhancement circuit C derives an improved image value for each pixel, I'(i,j), which is related to a combination of the pixel value, I(i,j) and the corresponding mean pixel neighborhood value $\bar{I}(i,j)$ weighted in accordance with the transfer function G(i,j), specifically:

$$I'(i,j) = G(i,j)[I(i,j) - \bar{I}(i,j)] + \bar{I}(i,j) \quad (11).$$

A subtraction means 54 subtractively combines pixel by pixel the pixel intensity I(i,j) and the mean neighborhood value $\bar{I}(i,j)$. A multiplying means 56 multiplies the difference between the pixel and mean neighborhood values by the corresponding transfer function value G(i,j). An adding means 58 additively combines the mean neighborhood value and the transfer function product pixel by pixel. Each resultant enhanced/filtered pixel value I'(i,j) is stored at the corresponding pixel of a second or improved image memory 60. The improved image may be displayed on a video monitor 62 or other display means. Optionally, the image may be stored on magnetic tape or disk subject to further processing, or the like.

With continuing reference to FIG. 1 and further reference to FIG. 4, if a portion of the resultant image is to be enlarged for closer review, the zoom circuit D is activated. In the enlarged or zoomed image, there are more pixels than there are pixel values in the original image from the improved image memory 60. A subregion loading means 70 loads the pixel values from the region to be enlarged in the original image into corresponding pixels which are distributed uniformly over a zoom memory 72. If the region of the original image is to be doubled in width and height, then there are empty pixels, note pixels 74 and 76, between each filled pixel of each column. Analogously, between each row of filled pixels, the zoom memory 72 has an empty row of pixels, note rows 78 and 80. It is to be appreciated that if the region of interest is to be more enlarged, there will be more intervening unfilled pixels. If the region of interest is to be less enlarged, there will be fewer unfilled pixels. The zoom means D derives pixel values for each empty pixel in the zoom memory 72 from the pixel values loaded from the original image 60 by the loading means 70. More specifically, the zoom circuit D fills each empty pixel with a value that is equal to the sum of an average of neighboring pixel values and a fractal value F.

An averaging means 82 averages the pixel values of the filled pixels which neighbor or are contiguous to each empty pixel. For example, empty pixel 74 is filled with the average of the neighboring pixel values (i,j) and (i+1,j), i.e. $[I(i,j)+I(i+1,j)]/2$. Optionally, the average may be based on other average or mean value algorithms.

An image may be thought of as an intensity surface, where each pixel intensity value represents height (z-coordinate) above the corresponding pixel location on x-y plane.

The fractal value F is a function of the roughness or variation in neighboring values and random numbers. A roughness means 84 calculates a roughness factor R for each empty pixel based on the self-similarity factors of neighboring filled pixels. Preferably, the roughness factor is set equal to a power of the average self-similarity factor of the adjoining filled pixels. For pixel 74, the roughness factor is:

$$R = k2^{-[H(i,j)+H(i+1,j)]/2} \quad (12),$$

where k is a constant selected in accordance with system parameters, the video monitor, and the desired

characteristics of the zoomed image. Again, the exponent may be based on the nearest neighbors and more distant neighboring pixels. A multiplying means 86 multiplies the roughness factor R for each interpolated pixel by a random number generated by random number generator 88 to generate the fractal value F. An adding means 90 combines the fractal value and the average value to produce the interpolated pixel value, i.e.:

$$I(\text{inter})=[I(i,j)+I(i+1,j)]/2+R\cdot(\text{random \#}) \quad (13).$$

The interpolated value for pixel 78 may be calculated analogously based on filled pixel values I(i,j) and I(i,j+1). The interpolated value for a pixel 92 at the intersection of empty rows and columns may be determined from the intensities and self-similarity factors from the four nearest neighbor pixels (i,j), (i+1,j), (i,j+1), and (i+1, j+1). Alternately, the pixel value 92 may be interpolated analogously from the four nearest interpolated pixel values or from a combination of the four nearest interpolated pixel values and the four nearest filled pixel values.

The invention has been described with reference to the preferred embodiments. Obviously, modifications and alterations will occur to others upon reading and understanding the preceding detailed description. It is intended that the invention be construed as including all such modifications and alterations insofar as they come within the scope of the appended claims or the equivalents thereof.

Having thus described the preferred embodiment, the invention is now claimed to be:

1. An apparatus for generating medical diagnostic image representations, the apparatus comprising:

a medical diagnostic apparatus for generating diagnostic data indicative of a preselected image region of a patient;

an imager for generating an electronic image representation from the diagnostic data, the electronic image representation including a pixel value corresponding to each pixel of a pixel array, each pixel value is indicative of an image property of a corresponding subregion of the image region of the patient;

a self-similarity value generating means for generating a self-similarity value corresponding to each pixel, the self-similarity value generating means being operatively connected with the imager to receive pixel values therefrom, each self-similarity value varying in accordance with a ratio of (i) a difference between the corresponding pixel value and a first set of pixel values contiguous to and surrounding the corresponding pixel value and (ii) the corresponding pixel value and a second set of pixel values contiguous to and surrounding the first set of pixel values;

an image improvement means for replacing each pixel value by a combination of the replaced pixel value and an average of surrounding pixel values, the combination being weighted in accordance with the corresponding self-similarity value, the image improvement means being operatively connected with the imager and the self-similarity value generating means.

2. An apparatus for generating medical diagnostic image representations, the apparatus comprising:

a medical diagnostic apparatus for generating medical diagnostic signals indicative of an image region of a patient;

an imager for transforming the medical diagnostic signals into electronic image representations, each electronic image representation including a pixel value corresponding to each pixel of a pixel array;

a loading means for loading each pixel value into uniformly distributed, spaced pixels of a zoom memory means such that the pixel values are stored in uniformly distributed filled pixels and empty pixels are disposed therebetween;

an interpolating means for generating a pixel value for each empty pixel, the interpolating means including:

an averaging means for generating an average of neighboring pixel values;

a random number generator for generating a random number;

a self-similarity value generating means for generating a self-similarity value corresponding to each pixel, the self-similarity value generating means being operatively connected with the imager to receive pixel values therefrom, each self-similarity value varying in accordance with a ratio of differences between the corresponding pixel value and pixel values in at least two rings contiguous to the corresponding pixel value;

a weighting means for weighting the random number in accordance with the self-similarity value, the weighting means being operatively connected with the self-similarity value generating means and the random number generator; and,

combining means for combining the pixel value average from the averaging means and the weighted random number from the weighting means, the combining means being operatively connected with the averaging means to receive the average value therefrom, with the weighting means for receiving the weighted random number therefrom, and with the zoom memory means for storing the combined average pixel value and weighted random number at the corresponding empty pixel thereof.

3. A method of medical diagnostic imaging comprising:

converting medical diagnostic data into an electronic image representation which includes an array of pixel values;

generating a self-similarity value corresponding to each pixel value of the electronic image representation, each self-similarity value being a dimensionless value which varies with a degree of irregularity among pixel values in a neighborhood of the corresponding pixel value;

providing an improved electronic image representation by replacing each pixel value by a combination of the replaced pixel value and an average of surrounding pixel values, the combination being weighted in accordance with the corresponding self-similarity value.

4. The method as set forth in claim 3 wherein the self-similarity value generating step includes:

determining a difference between the corresponding pixel value and each pixel value within a nearest neighbor first ring and averaging the first ring differences;

determining a difference between the corresponding pixel value and pixel values in a next nearest neighboring second ring and averaging the second ring differences.

5. The method as set forth in claim 4 wherein the self-similarity value generating step further includes determining a logarithm of a ratio of the first ring and second ring differences.

6. The method as set forth in claim 5 further including multiplying the logarithm by a constant selected in accordance with the diameter of the rings.

7. The method as set forth in claim 5 wherein the replacing step further includes determining a transfer function for each pixel value from the corresponding self-similarity value and wherein the combination is weighted by multiplying the combination by the transfer function.

8. The method as set forth in claim 7 wherein the transfer function is inversely proportional to the self-similarity value for at least a selected range of self-similarity values.

9. The method as set forth in claim 7 wherein the transfer function is set equal to a first constant when the self-similarity value exceeds a preselected value and the transfer function is set equal to a second preselected constant when the self-similarity value is equal to or less than the preselected value.

10. The method as set forth in claim 4 further including enlarging a selected portion of the electronic image representation, the enlarging step including distributing pixel values from the selected image representation portion substantially uniformly among available pixels of an enlarged image electronic representation such that a fraction of the enlarged image representation pixels are filled by the transferred pixel values and empty pixels are defined therebetween and interpolating neighboring filled pixel values to derive pixel values for the empty pixels.

11. The method as set forth in claim 10 wherein the interpolating step includes:

for each empty pixel, averaging pixel values from neighboring filled pixels; and,

adding to the averaged pixel valves a fractal value which varies in accordance with a random number and the self-similarity value.

12. The method as set forth in claim 11 wherein the fractal value varies in proportion to a constant raised to a power of an average of the self-similarity values corresponding to the neighboring filled pixels.

13. A method of improving an image comprising:

converting data into an electronic image representation which includes an array of pixel values;

determining a first difference between the corresponding pixel value and each pixel value within a first surrounding ring and averaging the first ring differences;

determining a difference between the corresponding pixel value and each pixel value in a second surrounding ring and averaging the second ring differences;

generating a self-similarity value corresponding to each pixel value of the electronic image representation in accordance with a difference between the first and second difference averages corresponding to the same pixel value;

providing an improved electronic image representation by replacing each pixel value by a combination of the replaced pixel value, pixel values of pixels surrounding the replaced pixel value, and the self-similarity value corresponding to the replaced pixel value.

14. A method of medical diagnostic imaging comprising:

converting medical diagnostic data into an electronic image representation which includes an array of pixels, each pixel having a pixel value;

generating a self-similarity value corresponding to each pixel;

enlarging a selected portion of the electronic image representation, the enlarging step including distributing pixel values of the selected image representation portion substantially uniformly among pixels of an enlarged image electronic representation such that a fraction of the enlarged image representation pixels are filled by the transferred pixel values and empty pixels are defined therebetween;

for each empty pixel, averaging pixel values from neighboring filled pixels;

for each empty pixel deriving a fractal value which varies in accordance with a random number and the self-similarity values of the neighboring filled pixels; and,

combining the fractal value and the neighboring pixel value average that corresponds to the same empty pixel.

15. The method as set forth in claim 14 wherein the fractal value varies in proportion to a constant raised to a power of an average of the self-similarity values corresponding to the neighboring filled pixels.

16. The method as set forth in claim 15 wherein each self-similarity value varies in accordance with a ratio of variations between the corresponding pixel value and pixel values in at least two surrounding rings.

17. The method as set forth in claim 15 wherein the self-similarity value generating step includes:

determining a difference between the corresponding pixel value and each pixel value within a first surrounding ring and averaging the first ring differences;

determining a difference between the corresponding pixel value and each pixel value in a second surrounding ring and averaging the second ring differences.

18. The method as set forth in claim 17 wherein the self-similarity value generating step further includes determining a logarithm corresponding to a ratio of the first ring and second ring differences.

19. The method of claim 18 including providing an improved electronic image representation by replacing each pixel value by a combination of the pixel value and an average of surrounding pixel values, each combination being weighted in accordance with the corresponding self-similarity value.

20. A method of enlarging a selected portion of an image representation comprising:

converting data into an electronic image representation which includes an array of pixels, each pixel having a pixel value;

generating a self-similarity value corresponding to each pixel from pixel values of adjacent pixels;

distributing pixel values of the selected image representation portion substantially uniformly among pixels of an enlarged image electronic representation such that a fraction of the enlarged image representation pixels are filled by the transferred

pixel values and empty pixels are defined therebetween;

for each empty pixel, averaging pixel values from neighboring filled pixels;

for each empty pixel, deriving a fractal value which varies in accordance with a random number and the self-similarity values of adjacent pixels; and,

combining the fractal value and the neighboring pixel value average that corresponds to the same empty pixel.

* * * * *

TAB G

IEEE TRANSACTIONS ON

PATTERN ANALYSIS AND MACHINE INTELLIGENCE

NOVEMBER 1981 VOLUME PAMI-3 NUMBER 6 (ISSN 0162-8828)

PUBLISHED BY THE IEEE COMPUTER SOCIETY

In Cooperation With

Aerospace and Electronic Systems Society
Control Systems Society
Engineering in Medicine and Biology Society
Information Theory Group
Sonics and Ultrasonics Group
Systems, Man, and Cybernetics Society

PAPERS

On the Cellular Convexity of Complexes ..... C. E. Kim 617
Parallel Region Property Computation by Active Quadtree Networks .... T. Dubitzki, A. Y. Wu, and A. Rosenfeld 626
Semantic Description of Aerial Images Using Stochastic Labeling ..... O. D. Faugeras and K. E. Price 633
Optimal Solution of Linear Inequalities with Applications to Pattern Recognition .. D. C. Clark and R. C. Gonzalez 643
Real-Time Adaptive Contrast Enhancement ..... P. M. Narendra and R. C. Fitch 655
Determining Surface Orientations of Specular Surfaces by Using the Photometric Stereo Method ..... K. Ikeuchi 661

CORRESPONDENCE

Some Results in the Processing of the Holy Shroud of Turin ..... G. Tamburelli 670
A Simple Contour Matching Algorithm ..... T. W. Sze and Y. H. Yang 676
Image Smoothing Based on Neighbor Linking ..... J. O. Eklundh and A. Rosenfeld 679
Computing Perimeters of Regions in Images Represented by Quadtrees ..... H. Samet 683
Image Approximation from Gray Scale "Medial Axes" ..... S. Wang, A. Y. Wu, and A. Rosenfeld 687
Comments on "Nosing Around the Neighborhood: A New System Structure and Classification Rule for Recognition in Partially Exposed Environments" ..... P. A. Devijver 696
An Efficient Two-Dimensional FFT Algorithm ..... L. R. Johnson and A. K. Jain 698
Evaluation of Projection Algorithms ..... G. Biswas, A. K. Jain, and R. C. Dubes 701
Elastic Matching of Line Drawings ..... D. J. Burr 708

BOOK REVIEWS

Remote Sensing: The Quantitative Approach—P. H. Swain and S. M. Davis, Eds. .... Reviewed by W. C. Kennard 713
Pattern-Directed Inference Systems—D. A. Waterman and F. Hayes-Roth, Eds. ..... Reviewed by M. Selfridge 714

1981 INDEX .. .. Follows page 715

IEEE COMPUTER SOCIETY

The Computer Society is an association of people with professional interest in the field of computers. All members of the IEEE are eligible for membership in the Society upon payment of the annual Society membership fee of $8.00. Members of certain professional societies and other computer professionals are also eligible to be members of the Computer Society. For information on joining, write to IEEE Computer Society, 1109 Spring Street, Suite 202, Silver Spring, MD 20910.

* Deceased

[1] Acting President
O. N. Garcia
Dept. of Comput. Sci. & Eng.
University of South Florida
Tampa, FL 33620

[2] Executive Committee Member

[3] PO Box 639, Silver Spring, MD 20901

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE is published bimonthly by The Institute of Electrical and Electronics Engineers, Inc. Headquarters: 345 East 47 Street, New York, NY 10017. Responsibility for the contents rests upon the authors and not upon the IEEE, the Society, or its members. IEEE Service Center (for orders, subscriptions, address changes): 445 Hoes Lane, Piscataway, NJ 08854. Telephone: Headquarters 212-644 + extension: Information-7900, General Manager-7910, Controller-7748, Education Services-7860, Publishing Services-7560, Regional/Section Services-7751, Standards-7960, Technical Services-7890, IEEE Service Center 201-981-0060, Washington Office/Professional Services 202-785-0017. NY Telecopier: 212-752-4929. Telex: 236-411 (International messages only). Individual copies: IEEE members $5.00 (first copy only), nonmembers $10.00 per copy. Annual subscription price: IEEE members, dues plus Society fee. Price for nonmembers on request. Available in microfiche and microfilm. Printed in U.S.A. Second-class postage paid at New York, NY and at additional mailing offices.

[5] R. E. Warmack and R. C. Gonzalez, "An algorithm for the optimal solution of linear inequalities and its application to pattern recognition," *IEEE Trans. Comput.*, vol. C-22, no. 12, pp. 1065–1075, 1973.
[6] A. Miyake and S. Shinmura, "An algorithm for the optimal linear discriminant functions," in *Proc. Int. Conf. Cybern. Soc.*, vol. 2, 1978, pp. 1447–1450.
[7] S. Shinmura and A. Miyake, "Optimal linear discriminant functions and their application," in *Proc. Comput. Software Appl. Conf.*, Chicago, IL, Nov. 6–8, 1979, pp. 167–172.
[8] A. Miyake, "Mathematical aspects of optimal linear discriminant functions," in *Proc. Comput. Software Appl. Conf.*, Chicago, IL, Nov. 6–8, 1979, pp. 161–166.
[9] H. Mengert, "Solution of linear inequalities," *IEEE Trans. Comput.*, vol. C-19, no. 2, pp. 124–131, 1970.
[10] F. W. Smith, "Pattern classifier design by linear programming," *IEEE Trans. Comput.*, vol. C-17, no. 4, pp. 367–372, 1968.
[11] R. G. Grinold, "Comment on 'Pattern classifier design by linear programming,'" *IEEE Trans. Comput.*, vol. C-18, no. 4, pp. 378–379, 1969.
[12] E. W. Cheney, *Introduction to Approximation Theory*. New York: McGraw-Hill, 1966.
[13] V. Klee, "What is a convex set," *Amer. Math. Mon.*, vol. 78, pp. 617–631, June–July, 1971.
[14] R. T. Rockafellar, *Convex Analysis*. Princeton, NJ: Princeton Univ. Press, 1970.
[15] T. M. Cover, "Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition," *IEEE Trans. Electron. Comput.*, vol. EC-14, no. 6, pp. 326–334, 1965.
[16] J. K. Bryan and D. L. Tebbe, "Generation of multivariate Gaussian data," Dep. Elec. Eng., Univ. Missouri, Colombia, Tech. Rep., Apr. 1970.
[17] G. N. Wassel, "Training a linear classifier to optimize the error probability," Ph.D. dissertation, Univ. California, Irvine, Dec. 1972.
[18] H. Do-Tu and M. Installe, "Learning algorithms for nonparametric solution to the minimum-error classification problem," *IEEE Trans. Comput.*, vol. C-27, pp. 648–659, 1978.

D. C. Clark received the B.S. degree in mathematics from the California Institute of Technology, Pasadena, in 1963, the Ph.D. degree in mathematics from Stanford University, Stanford, CA, in 1968, and the M.S. degree in computer science from the University of Tennessee, Knoxville, in 1980.

He has taught in the Departments of Mathematics at Rutgers University, Newark, NJ, and the University of Puerto Rico, Mayaguez. Currently, he is with the Los Angeles office of Pattern Analysis and Recognition Corporation.

R. C. Gonzalez (S'65–M'70) was born in Havana, Cuba, on August 26, 1942. He received the B.S.E.E. degree from the University of Miami, Coral Gables, FL, in 1965 and the M.E. and Ph.D. degrees in electrical engineering from the University of Florida, Gainesville, in 1967 and 1970, respectively.

He has been affiliated with the GT&E Corporation, the Center for Information Research at the University of Florida, NASA, and is presently IBM Professor of Electrical Engineering and Computer Science at the University of Tennessee, Knoxville. He is a frequent consultant to industry and government in the areas of pattern recognition, image processing, and machine learning. He received the 1978 UTK Chancellor's Research Scholar Award, the 1980 Magnavox Engineering Professor Award, and the 1980 M. E. Brooks Distinguished Professor Award for his work in these fields. He is coauthor of the books *Pattern Recognition Principles*, *Digital Image Processing*, and *Syntactic Pattern Recognition: An Introduction*, all published by Addison-Wesley. He is an Associate Editor for the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS and the *International Journal of Computer and Information Sciences*.

Dr. Gonzalez is a member of several professional and honorary societies, including Tau Beta Pi, Phi Kappa Phi, Eta Kappa Nu, and Sigma Xi.

Real-Time Adaptive Contrast Enhancement

PATRENAHALLI M. NARENDRA, MEMBER, IEEE, AND ROBERT C. FITCH, MEMBER, IEEE

Abstract—A recursive filter approach is introduced to simplify real-time implementation of an adaptive contrast enhancement scheme for imaging sensors. With this scheme, even scenes possessing large global dynamic ranges (>40 dB) can be accommodated by the limited dynamic range (20 dB) of a display without losing the local contrast essential for image interpretation. This paper describes the recursive filter implementation of the local area contrast enhancement scheme using charge-coupled devices and the resultant real-time hardware capable of processing standard 525 and 875 line TV compatible video (from vidicons, videotape recorders, etc). Several examples from video imagery are included to demonstrate its effectiveness.

Index Terms—AGC, contrast enhancement, digital filter, dynamic range compression, FLIR, image enhancement, real-time processing, signal processing architectures, TV, video.

Manuscript received March 10, 1980; revised February 6, 1981. This work was supported in part by the U.S. Army Night Vision and Electro Optics Laboratory under Contract DAAG53-76-C-0195.

The authors are with the Systems and Research Center, Honeywell, Inc., Minneapolis, MN 55413.

I. INTRODUCTION

THE SCENE dynamic ranges encountered by imaging sensors can be much higher than the CRT display luminance range (~20 dB). Moreover, the scene intensity extremes are changing in time. It is obvious that some form of global automatic gain/bias control (AGC) system can be done by a linear scaling of the intensities governed by the scene extrema or the global scene variance.

But mere global gain and bias control does not suffice when the scene has a large dynamic range. When the high dynamic range scene information is squeezed into the limited range of a typical display, low contrast local details are not perceived. Therefore, in addition to the global gain and bias control we need local area contrast enhancement—to enhance the local contrast while compressing the global scene dynamic range presented to the display. Previous work in this direction included Ketcham [1] and Harris [2]. Both are essentially high

0162-8828/81/1100-0655$00.75

frequency emphasis schemes, and use local area statistics (mean and variance) computed on a sliding window to control the contrast and bias level at each image point. Note that these are local transformation schemes as distinct from global gray scale transformations also suggested for enhancing display contrast [3], [4].

Our present approach replaces the complex computations of the local area statistics with simple two-dimensional recursive low-pass filters. This enables a very simple real-time implementation at video rates. CCD line delays are used to provide the time delays needed in the recursive filter. This in turn results in a sampled analog structure that uses CCD line delays, analog multipliers, and adders in the 10–30 MHz rates encountered in real-time video applications.

In Sections II and III we describe the basic local area contrast enhancement (LACE) scheme, discuss various alternative implementations of the local area statistics computation, and discuss the application of the recursive filter to greatly simplify these operations. The description of the simple hardware implementing this approach follows in Section IV. We include several examples of videotaped imagery enhanced by this real-time hardware to demonstrate its effectiveness.

II. The Local Area Contrast Enhancement (LACE) Scheme

Fig. 1 is a one-dimensional illustration of a scene which can be seen as a slowly varying background envelope. Superimposed upon it is higher frequency local variation representing scene detail and other information cues. The problem with real-life imagery is that the gain has to be made small enough to squeeze the wide dynamic range of the scene into the small (20:1) luminance range of the display. Then the *local* contrast (i.e., difference in intensities) falls below the contrast sensitivity threshold ΔI of the human eye and will not be perceived. The contrast sensitivity threshold ΔI is a function of both the local-area size and the average local intensity. For a given local-area size, the contrast sensitivity obeys Weber's law (i.e., is proportional to the intensity, $\Delta I/I = C_s$, a constant). This contrast C_s can vary from less than 1 percent for large areas to infinity for very small local areas.

Local area contrast enhancement (LACE) schemes operate under the premise that the large slowly varying intensity excursions can be reduced without degrading the information in the scene. The local contrast can then be enhanced (by increasing the local gain) without exceeding the dynamic range of the display. Fig. 1(b) illustrates this; the low frequency envelope is brought closer to the global mean while the higher frequency local variations are now amplified above the contrast sensitivity threshold ΔI. This does not significantly degrade the information content of the image since the relative brightness of distant areas within the scene contributes little to discrimination.

A LACE formulation that addresses the above visual psychophysical considerations is derived here. It performs the following functions:

- vary the local average brightness (bias) so that overall dynamic range of scene is compressed;
- enhance local variations above the contrast sensitivity threshold of the human eye; and

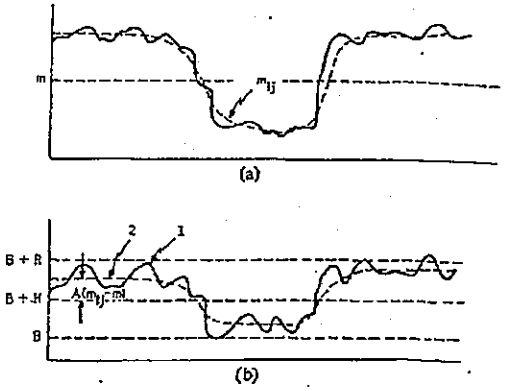


Fig. 1. (a) Original one-dimensional scene example. (b) LACE applied: 1) high frequency component and 2) low frequency component.

- automatically fit the intensity extremes in the enhanced scene to the display limits.

A functional description of the algorithm is shown in Fig. 2. The image intensity at each point is transformed based on local area statistics—the local mean M_{ij} and the local standard deviation σ_{ij} computed on a local area surrounding the point. The transformed intensity is then

$$\hat{I}_{ij} = G_{ij}(I_{ij} - M_{ij}) + M_{ij} \tag{1}$$

where the local gain is

$$G_{ij} = \alpha \frac{M}{\sigma_{ij}}, \quad 0 < \alpha < 1$$

where M is the global mean.

The local area mean is first subtracted from the image at every point. A variable gain is applied to the difference to amplify the local variations. A portion of the local mean M_{ij} is then added back to restore the subjective quality of image. The local gain G_{ij} is itself locally adaptive, being proportional to M_{ij}, to satisfy psychovisual considerations (Weber's law); and inversely proportional to σ_{ij}, so that areas with small local variance receive larger gain. To prevent the gain from being inordinately large in areas with large mean and small standard deviation, the local gain is actually controlled as in Fig. 3.

Note that this formulation is virtually identical to those of [1] and [2]. The main contribution of this paper is in an efficient recursive architecture for the implementation of this scheme, discussed in the next section.

III. Real-Time Implementation

Three alternate implementations have been considered to realize the basic scheme in Fig. 2. They differ mainly in the way the local area mean and standard deviation are computed in real time.

Nonrecursive Implementation

The most obvious implementation is to have an $M \times M$ sliding window using shift registers (line delays) to perform nonrecursive computation of the mean and standard deviation [1]. This requires simultaneous access to M lines—which implies an inordinate amount of line delays and high-speed addition (over $2M$ adds/pixel at 8–30M samples/s) at the video rates en-

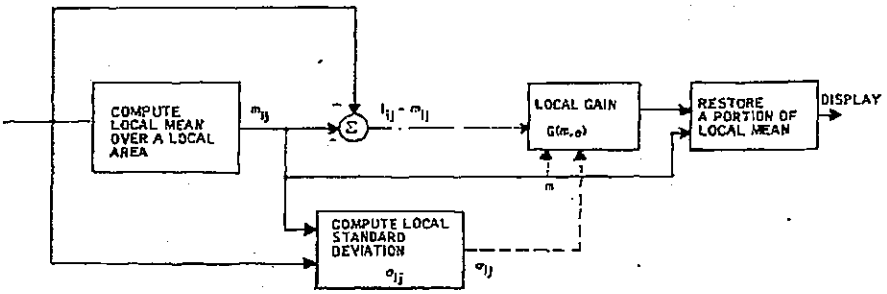


Fig. 2. Functional flow description of the local area contrast enhancement algorithm.

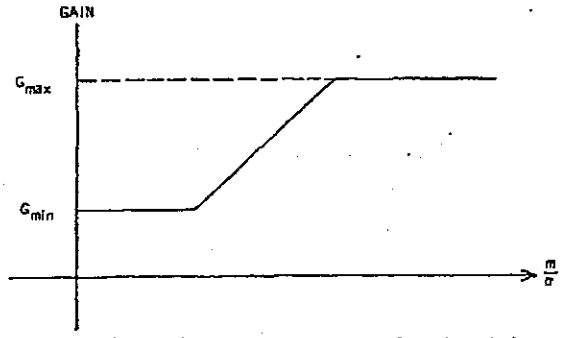


Fig. 3. Local area gain curve to prevent excessive gain variations.

countered in imaging sensors. A 15 X 15 window requires real-time buffering (in the shift registers) of 15 lines of video, and 225 adds/sample period.

Bilinear Interpolation

To reduce the hardware complexity of nonrecursive computation on a sliding window, an alternate approach was computer simulated. Here, the local mean and standard deviation are computed on *nonoverlapping* local areas of size M and stored for each frame. The local mean and standard deviation at any point in the next frame are then estimated by a simpler bilinear interpolation. This approach is effective because it cuts down the mean and standard deviation computation to N^2 MAD's versus M^2N^2 (a saving by a factor of over 100) without any of the grid-like artifacts that would be inherent in other nonoverlapping window approaches. This is because the interpolated values of the mean standard deviation vary smoothly across the region boundaries.

This approach was found to be very efficient and effective in computer simulations. But its real-time implementation is still not as simple as we would like it to be. Although the number of multiplies and adds per pixel is greatly reduced, the storing, addressing, and updating of the local area statistics and the interpolation require a complex architecture.

Recursive Implementation

The recursive approach achieves a very simple structure suitable for real time implementation. It is obvious that the local mean and standard deviation computation is in fact nonrecursive low-pass filtering of the input and (input - local mean)^2 with a rect function. If we replace these nonrecursive low-pass filters by two-dimensional recursive low-pass filters, we stand to gain considerable simplicity in the resultant implementation.

Fig. 4 is a realization of the LACE scheme using linear recursive low-pass filters to estimate the local mean and standard deviation. Here we replace the local average function with an equivalent two-dimensional recursive low-pass filter. The local standard deviation σ_{ij} is approximated by a similarly low-passed version of the absolute difference between the image intensity I_{ij} and the local mean estimate M_{ij}.[1]

A two-dimensional separable first-order recursive low-pass filter is the basic building block in Fig. 4. This filter has a frequency response given by the product of the two one-dimensional filter responses:

$$|H(f_x, f_y)| = [1 + (f_x/f_c)^2]^{-1/2} [1 + (f_y/f_c)^2]^{-1/2} \quad (3)$$

where f_c is the 3 dB cutoff frequency of the low-pass filter. The equivalent sampled data filter has a two-dimensional Z transform,

$$H(z_1, z_2) = \frac{\gamma^2}{(1 - e^{-\gamma} z_1^{-1})(1 - e^{-\gamma} z_2^{-1})} \quad (4)$$

where

$$\gamma = 2\pi f_c / f_s.$$

Changing γ changes the effective size of the local area, i.e., the area over which the local mean averaging is done. The above separable filter can be realized using two distinct (but functionally equivalent) structures.

Nonseparable Implementation: In the Z domain, let the output be $Y(z_1, z_2)$ and the input be $X(z_1, z_2)$. Then,

$$\frac{Y(z_1, z_2)}{X(z_1, z_2)} = \frac{\gamma^2}{(1 - e^{-\gamma} z_1^{-1})(1 - e^{-\gamma} z_2^{-1})}. \quad (5)$$

This form, implemented directly, gives the recursive relation

$$y(m,n) = \gamma^2 x(m,n) + e^{-\gamma} y(m-1,n) + e^{-\gamma} y(m,n-1) - e^{-2\gamma} y(m-1,n-1) \quad (6)$$

where m is the row number and n is the column number in the image.

[1]Note that for a Gaussian distribution, $E|x - \mu| = \sqrt{2/\pi}\,\sigma$, where μ is the mean and σ is the standard deviation.

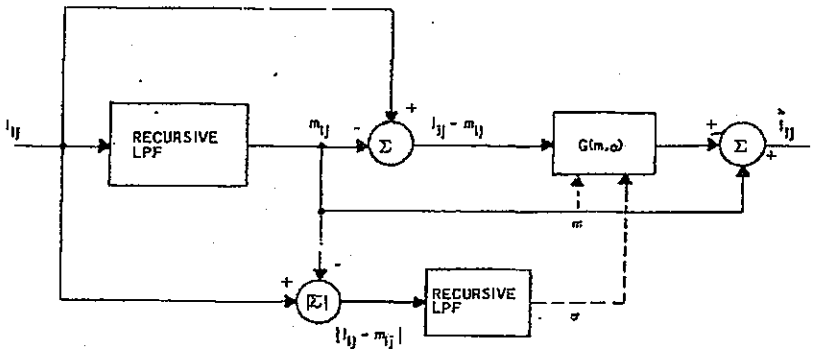


Fig. 4. LACE with linear recursive low-pass filters.

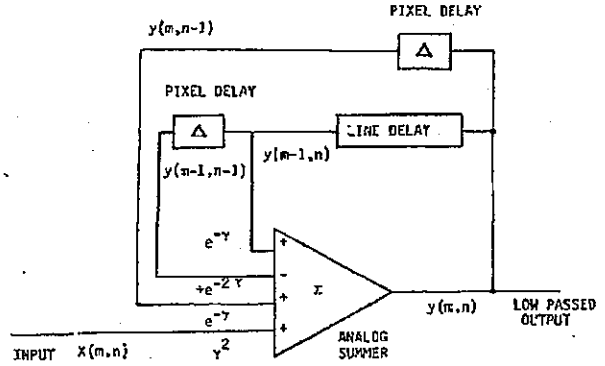
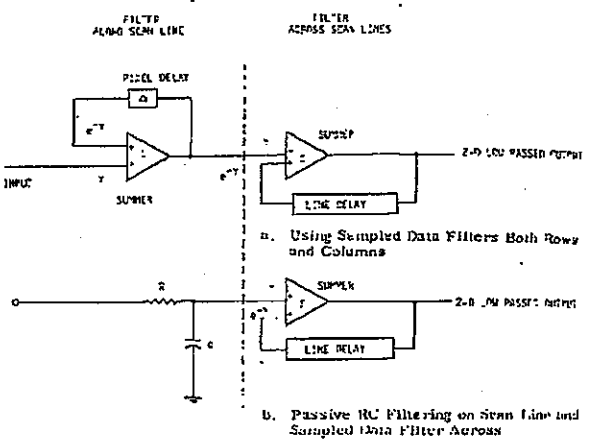


Fig. 5. Implementation of two-dimensional recursive low-pass filter (Approach 1).



a. Using Sampled Data Filters Both Rows and Columns

b. Passive RC Filtering on Scan Line and Sampled Data Filter Across

Fig. 6. Separable implementation of two-dimensional low-pass filter (Approach 2).

The schematic for implementing this realization on a real-time video stream is shown in Fig. 5. We see that just one line delay and the two pixel delays are necessary to perform the filtering.

Separable Implementation: Define a new intermediate variable

$$W(z_1, z_2) = \frac{\gamma}{(1 - e^{-\gamma} z_1^{-1})} \cdot X(z_1, z_2). \tag{7}$$

Then,

$$\frac{Y(z_1, z_2)}{X(z_1, z_2)} = W(z_1, z_2) \cdot \frac{\gamma}{(1 - e^{-\gamma} z_2^{-1})}. \tag{8}$$

Therefore,

$$y(m, n) = \gamma w(m, n) + e^{-\gamma} y(m - 1, n) \tag{9}$$

and

$$w(m, n) = \gamma x(m, n) + e^{-\gamma} w(m, n - 1). \tag{10}$$

Thus we break up the two-dimensional filter into two one-dimensional filters in cascade. Fig. 6 shows this filter realized in the above manner. The output of this filter is exactly equivalent to that of Fig. 5 because the transfer function is separable. This realization has certain advantages over the nonseparable implementation as described below.

The parameter γ controls the effective size of the local area. For local area (impulse response area) sizes desired in LACE, γ is in the range 0.1–0.3, which makes $e^{-\gamma} \cong 1 - \gamma$, in the range 0.9–0.7. In the nonseparable realization, the smallest weight of the summing amplifier, $\gamma^2 = 0.1$–0.01, is much smaller than the largest weight $1 - \gamma = 0.7$–0.9. The coefficient precision needed in the summer in Fig. 5 is therefore very stringent. On the other hand, the coefficient ranges in the separable formulation is much smaller: $\gamma = 0.1$–0.3 for the smallest weight and $1 - \gamma = 0.9$–0.7 for the largest. Therefore, the separable structure is less susceptible to charge-coupled device (CCD) noise and amplifier gain variation than is the corresponding nonseparable structure. In a corresponding digital implementation as well, the separable realization would require a smaller precision adder.

The separable structure has an additional advantage in an analog implementation. The first low-pass filter (along the scan direction) can be easily implemented in a passive RC first-order circuit, as shown in Fig. 6(b). At video frequencies, the required resistor (R) and capacitor (C) values are very reasonable. This eliminates the need for the single pixel delay for the

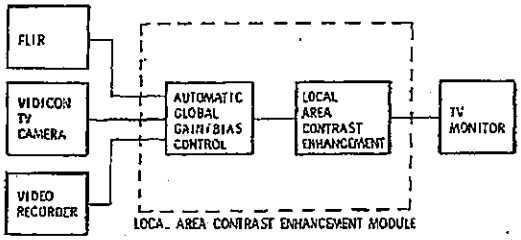


Fig. 7. Local area contrast enhancement module configuration.

first filter. However, the second low-pass filter in the vertical direction is still a sampled data filter as before, requiring one line delay.

We see that the first-order recursive filter above can achieve equivalent size local averaging as a nonrecursive window with just one line delay and two weighting coefficients. The impulse response is an exponential, $h(m,n) = e^{-(m+n)\gamma}, m, n \geq 0$ controls the size of the impulse response. Making γ smaller makes the impulse response wider.

In the hardware description that follows, we used $\gamma = 0.1$. Note also that since the impulse response of the recursive filter is causal, the local area mean at a point evaluated by the filter is a function only of the upper left quadrant centered at that point. Therefore, the resultant filter is not zero-phase as a nonrecursive filter with a sliding window would be. Despite this, in the simulations we have performed and in the results of the real-time hardware, we observed no undue distortion caused by the nonzero phase of the recursive filter. Also since the filter is causal, we do not have to delay the input video to be in phase with the local mean before subtracting the local mean to Fig. 4. These advantages outweigh the nonzero phase property of the filter.

IV. Local Area Contrast Enhancement Hardware

The local area contrast enhancement uses two identical two-dimensional recursive filters—one to compute the local mean and the other to compute the local standard deviation as the schematic in Fig. 4 showed. The separable realization of the filter in Fig. 6, being the simplest, was chosen for hardware implementation. Commercially available CCD video line delays were used in the two recursive filters. All the remaining circuitry to realize the various summing and variable gain functions in Fig. 4, employ conventional high bandwidth analog op amps and multipliers. This breadboard was designed and built entirely with off-the-shelf components. Inspite of this constraint, the simplicity achieved by the recursive sampled analog filter approach resulted in a compact (6 X 6 in card) implementation of the entire LACE scheme.

Fig. 7 shows the general configurations in which this hardware can be used. It can receive 525 line or 875 line TV compatible video from any source—an FLIR, videotape recorder, video disk recorder, vidicon, etc. The output is a composite video signal capable of driving a 75 Ω load.

The input to the LACE unit is first automatically scaled between 0 and 1 V by the global gain/bias control unit using the frame video mean and standard deviation. This adjusts the contrast and brightness on a sync separated video input from a video disk, tape, camera, etc., so that effective use of the CCD dynamic ranges is made in the subsequent stages.

This module has been extensively tested with real-time videotaped 525 line imagery (at 30 frames/s). A few examples from this evaluation are reproduced in Fig. 8. These are photographs taken off the display before and after enhancement.

In the interests of a fair trial, the contrast and brightness levels on the monitors were not changed for the two conditions. In both cases, the video into the display received global scaling (i.e., had the same global extrema). This is in fact evident in the photographs themselves—the *global* contrast and average brightness remain the same before and after enhancement.

The most dramatic improvement in local area contrast can be seen in the FLIR images in Fig. 8. The large dark areas consist of smoke which tends to drive the video blacker than black on a conventional display, obscuring surrounding details. Local area contrast enhancement expands the contrast in these local areas so that the background and the objects suppressed by the smoke are now clearly visible. Note that other areas in the image with adequate local contrast are affected to a lesser degree. This is as it should be—we do not want to boost the local contrast (high frequency information) in already "contrasty" areas. Doing so magnifies the scene extrema and therefore reduces the relative global dynamic range available elsewhere.

As an added benefit, the enhanced images have crisper detail because local area contrast enhancement is in fact high frequency emphasis. This high frequency gain is adaptive—being greater in local areas with small local standard deviation.

V. Summary and Discussion

Conventional schemes for local area video contrast stretching suggest complex architectures for real-time implementations which require nonrecursive computation of the local area statistics on a sliding window. The size of the window dictates the hardware complexity, and window sizes between 10 X 10 and 15 X 15 pixels are usually considered optimal. A 15 X 15 window requires simultaneous access to 15 lines of video (14 shift registers or line delays) and 225 adds/pixel at 10-20M pixels/s.

The two-dimensional recursive filter approach in this paper requires just one line delay and one summer (adder) to accomplish the same function. Moreover, the effective size of the local area can be changed simply by changing the filter weights.

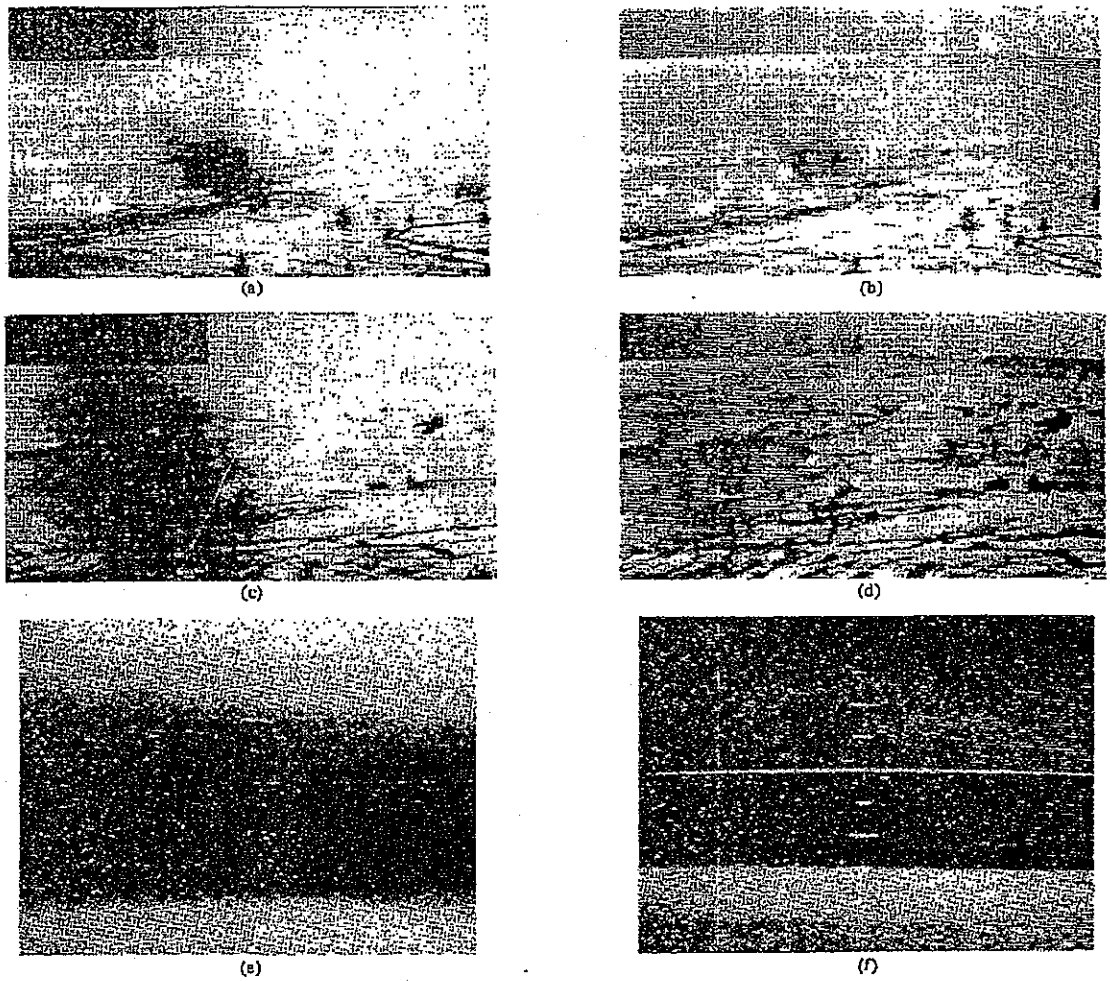


Fig. 8. Results of real-time processing of FLIR video through LACE module.

To change the local are in a nonrecursive sliding window, we would have to change the filter size, i.e., the entire filter structure.

The recursive filter approach would result in much greater simplicity (compared to the nonrecursive approach) in a digital implementation as well. We chose to implement it with sampled analog hardware using CCD line delays, because this approach eliminates high-speed A/D converters, multipliers, and adders at the 10–20 MHz rates encountered in 525 and 875 line video applications. It is interesting to note that the high frequency video does not pass through the CCD line delays in Fig. 6. Therefore, it is possible to use this circuitry without modifications beyond 30 MHz while keeping the CCD clock rates below 15 MHz. This is a boon since commercially available CCD line delays are limited to a 15 MHz clock frequency, but full resolution 875 line video can have bandwidth in excess of 30 MHz. A nonrecursive digital approach would be hard to put to realize these speeds.

This simple hardware occupies only one 6 X 6 in card although built entirely of off-the-shelf components. This image enhancement module has demonstrated the utility of CCD based sampled analog filtering techniques for simple hardware realizations of real-time image enhancement functions.

ACKNOWLEDGMENT

The authors wish to thank J. Dehne of the Night Vision and Electro Optics Laboratory and Dr. M. Geokezas of Honeywell, Inc. for their encouragement and support during this effort. Discussions with Drs. D. H. Tack and J. D. Joseph of Honeywell were very helpful.

REFERENCES

[1] D. J. Ketcham, "Real-time image enhancement techniques," in *Proc. Soc. Photo-Optical Instrumen. Eng.*, vol. 74 (Image Processing), Feb. 1976, pp. 120–125.
[2] J. L. Harris, "Constant variance enhancement–A digital processing technique," *Appl. Optics*, vol. 16, pp. 1268–1271, May 1977.
[3] R. C. Gonzalez and B. A. Fitles, "Grey level transformations for interactive image enhancement," *Mechanism and Machine Theory*, vol. 2, pp. 111–122, 1977.

[4] R. E. Woods and R. C. Gonzalez, "Real time digital image enhancement," *Proc. IEEE*, to be published.

Patrenahalli M. Narendra (S'73–M'76) was born in Jog Falls, India, on March 16, 1951. He received the Bachelor of Engineering degree in electronics from Bangalore University in 1971 and the M.S. and Ph.D. degrees in electrical engineering from Purdue University, West Lafayette, IN, in 1973 and 1975, respectively.

He has been with Honeywell Systems and Research Center, Minneapolis, MN, since 1976, where he is presently Staff Research Scientist in the Signal and Image Processing Section. His current research interests span dynamic scene analysis, symbolic image processing, and VLSI device architectures for real-time implementation of image processing systems. He is the author of several publications in his areas of interest. In 1979, he won the H. W. Sweatt Award, the highest technical achievement award in Honeywell.

Dr. Narendra is a member of Phi Kappa Phi and the IEEE Computer, PAMI, and ASSP Societies.

Robert C. Fitch (S'73–M'77) was born on June 23, 1953. He received the B.S. degree in electrical engineering from Montana State University, Bozeman, in 1975 and the M.S. degree in electrical engineering from Colorado State University, Fort Collins, in 1977.

He is currently a Senior Research Scientist in the Signal and Image Processing Section at Honeywell Systems and Research Center, Minneapolis, MN, where he has worked since 1977. His interests there are in VLSI implementations for real-time image processing, automatic target recognition, and robotics.

Mr. Fitch is a member of Tau Beta Pi and Phi Kappa Phi.

Determining Surface Orientations of Specular Surfaces by Using the Photometric Stereo Method

KATSUSHI IKEUCHI, MEMBER, IEEE

Abstract—The orientation of patches on the surface of an object can be determined from multiple images taken with different illumination, but from the same viewing position. The method, referred to as photometric stereo, can be implemented using table lookup based on numerical inversion of reflectance maps. Here we concentrate on objects with specularly reflecting surfaces, since these are of importance in industrial applications. Previous methods, intended for diffusely reflecting surfaces, employed point source illumination, which is quite unsuitable in this case. Instead, we use a distributed light source obtained by uneven illumination of a diffusely reflecting planar surface. Experimental results are shown to verify analytic expressions obtained for a method employing three light source distributions.

Index Terms—Bin of bolts and nuts, distributed light source, glossy object, reflectance map, shape from shading, surface inspection.

Manuscript received November 29, 1979; revised February 26, 1981.

The author is with the Computer Vision Section, Electrotechnical Laboratory, Ministry of International Trade and Industry, Ibaraki 305, Japan.

I. Introduction

THIS paper addresses the problem of determining local surface orientation of specular materials from the intensity information under different illumination, but from the same viewing position. This method is referred to as photometric stereo and was first formulated by Woodham [2]. Here, we concentrate on objects with a specularly reflecting surface, since these are of importance in industrial applications. Previous methods [2], [11] intended for diffusely reflecting surfaces employed point source illumination, which is quite unsuitable in this case.

Historically, Horn solved the image intensity equations [11] in order to obtain an object shape from shading information. Horn used a method of characteristic strip expansion for solving the image intensity equation which is a nonlinear first-order partial differential equation. Horn then introduced the reflectance map [3] in order to refine the image intensity

0162-8828/81/1100-0661$00.75

TAB H

DIGITAL IMAGE PROCESSING

Second Edition

Rafael C. Gonzalez
Electrical Engineering Department
University of Tennessee
Knoxville
and
Perceptics Corporation
Knoxville, Tennessee

Paul Wintz
Consultant

ADDISON-WESLEY PUBLISHING COMPANY

Reading, Massachusetts · Menlo Park, California
Don Mills, Ontario · Wokingham, England
Amsterdam · Sydney · Singapore · Tokyo
Madrid · Bogotá · Santiago · San Juan

Library of Congress Cataloging-in-Publication Data

Gonzalez, Rafael C.
Digital image processing.

Bibliography: p.
Includes index.
1. Image processing—Digital techniques. I. Wintz, Paul A. II. Title.
TA1632.G66 1987 621.36'7 86-28759
ISBN 0-201-11026-1

Reprinted with corrections November, 1987

HH-MA-89

APPENDIX A

IMAGE DISPLAY SUBROUTINES

This appendix contains two FORTRAN subroutines used for displaying image data of size 64 × 64 on a single sheet of line-printer paper. As discussed in Appendix B, the images included in that appendix are coded with the characters 0 through 9 and *A* through *V* to represent 32 gray levels. Once these characters have been read into an array denoted by AR, they have to be converted to an integer array, called IA, whose values range from 0 through 31. This conversion is accomplished by Subroutine CONVRT(AR), which is listed below. Note that the only parameter passed to this subroutine is the character array AR, which must be set up as CHARACTER*1 AR(64, 64) in the calling program. The output of this subroutine is integer array IA, which resides as COMMON IA(64, 64), INTEGER*2 IA, in all programs.

```
      SUBROUTINE CONVRT(AR)
C
C
C        THIS SUBROUTINE CONVERTS CHARACTER DATA RANGING FROM 0
C   TO 9 AND A TO V STORED IN ARRAY "AR" TO INTEGER DATA RANGING
C   FROM 0 TO 31, REPRESENTING THE SHADES OF GRAY.  THE INTEGER
C   DATA IS STORED IN ARRAY "IA".
C
C      MAJOR VARIABLES:
C
C        IA  ---   INTEGER ARRAY CONTAINING VALUES RANGING FROM
C                  0 TO 31 REPRESENTING THE SHADES OF GRAY.
```

```
C         AR  ---   CHARACTER ARRAY CONTAINING VALUES RANGING
C                   FROM 0 THROUGH 9 AND A THROUGH V WHICH
C                   CORRESPOND TO THE THIRTY-TWO GRAY LEVELS.
C
C       SUBPROGRAMS CALLED:      NONE
C
C       WRITTEN BY:
C                            NABEEL W. H. SUFI
C                      ELECTRICAL ENGINEERING DEPT.,
C                    UNIVERSITY OF TENNESSEE, KNOXVILLE
C
        COMMON IA(64,64)
        INTEGER*2 IA
        INTEGER I,J
        CHARACTER*1 AR(64,64)
C
C     LOOP THROUGH EACH ELEMENT OF 64*64 ARRAY
C
        DO 20 I=1,64
          DO 40 J= 1,64
C
C     CHECK IF THE CHARACTER IS BETWEEN 0 AND 9
C
            IF ((AR(I,J).GE.'0').AND.(AR(I,J).LE.'9')) THEN
              IA(I,J)=ICHAR(AR(I,J))-ICHAR('0')
C
C     IF CHARACTER IS NOT BETWEEN 0 AND 9 THEN
C     IT MUST BE BETWEEN A AND V BY DEFAULT
C
            ELSE
              IA(I,J)=ICHAR(AR(I,J))-ICHAR('A')+10
            ENDIF
 40       CONTINUE
 20     CONTINUE
C
        RETURN
        END
```

The subroutine actually used to display images on a line printer is subroutii DSP, which converts the integer values in array IA to gray levels by overstrikir characters on a line printer, as discussed below. Subroutine DSP is called as follow

CALL DSP(NX,NY,LAW,IL,IH,NEG,LG)

The arguments are:

NX—Number of rows of IA to be printed; maximum NX is 64.
NY—Number of columns of IA to be printed; maximum NY is 64.
NX = NY = 64 a full page is output.
LAW—Gray-level scale translation variable.
LAW = 1: linear scale,
LAW = 2: square-root scale,

LAW = 3: logarithmic scale,
LAW = 4: "absorption" scale.
IL—minimum gray level in IA, calculated in the calling program.
IH—maximum gray level in IA, calculated in the calling program.
NEG—a value equal to 1 gives the normal image; a value equal to 0 gives the negative of the image.
LG—logical unit number for the line printer.

The characters used in the program to obtain the 32 gray levels are shown in Fig. A.1. The characters in a column, when over-printed, produce the gray level indicated.

```
MMMMMMMHHHHHXHXOZWMNOS=I*++=:-.-
WWWWWW###*++----      =  -   -
####OO+-
OOO
+

                                    Gray levels
```

Figure A.1 Over-print characters used to obtain 32 gray levels. The 32nd character is a blank.

```
      SUBROUTINE DSP(NX,NY,LAW,IL,IH,NEG,LG)
C
C     **LINE PRINTER IMAGE OUTPUT SUBROUTINE**
C
C     ADAPTED BY B. A. FITTES, ELECTRICAL ENG.
C     DEPT., UNIVERSITY OF TENNESSEE, FROM
C     A PROGRAM WRITTEN BY J. L. BLANKENSHIP,
C     INSTRUMENTATION AND CONTROLS DIV., OAK
C     RIDGE NATIONAL LABORATORY, OAK RIDGE,
C     TN.  THE OVERPRINT METHOD USED IS FROM
C     "CONSIDERATIONS FOR EFFICIENT PICTURE
C     OUTPUT VIA LINE PRINTER," BY P. HENDERSON
C     AND S. TANIMOTO, REPORT NO. 153, 1974,
C     COMPUTER SCIENCE LAB., ELEC. ENG. DEPT.,
C     PRINCETON UNIVERSITY.
      COMMON IA(64,64)
      INTEGER*2 IA
      INTEGER*2 IB(64,64),LEV(32),BLANK(5)
```

```
      LOGICAL*1 LINE(128,5),GRAY(32,5)
C     SPECIFY GRAY-LEVEL CHARACTERS
      DATA GRAY /6*'M',5*'H','X','H','X','O',
     1 'Z','W','M','N','O','S','=','I','*','+',
     2 '+','=',':','-','.',' ','-',' ',6*'W',3*'#',
     3 '*','+','+',4*'-',5*' ','=',' ',' ','-',
     4 3*' ','-',3*' ',4*'#','O','O','+','-',
     5 24*' ','O','O','O',29*' ','+',31*' '/
      GN=32.0
      FL=IL
      FH=IH
      IF ((FH-FL).GT.0.0) GO TO 100
      T=FL
      FL=FH
      FH=T
  100 RANGE=(FH-FL+1)/GN
      AA=(SQRT(FH)-SQRT(FL))/GN
      EE=(FH-FL)/ALOG(GN+1.0)
      T=AMAX1(FL,1.0)
      SS=-(1.0/GN)*ALOG(FH/T)
C
C
C     A VECTOR LEV IS COMPUTED NEXT.  THIS
C     VECTOR IS A SET OF BREAK POINTS USED TO
C     DETERMINE THE SCALED VALUE OF IA(I,J).
C     THE MATRIX IB IS THE RESULT OF SCALING
C     IA.  IB(I,J)=K IF IA(I,J) IS LESS THAN
C     LEV(K+1) BUT GREATER THAN OR EQUAL TO
C     LEV(K).
C
      DO 160 I=1,32
      GO TO (110,120,130,140),LAW
  110 FLEV=FL+(I-1)*RANGE+0.5
      GO TO 150
  120 FLEV=(SQRT(FL)+(I-1)*AA)**2+0.5
      GO TO 150
  130 FLEV=FL+EE*ALOG(FLOAT(I))+0.5
      GO TO 150
  140 FLEV=FH*EXP(SS*(GN-I))+0.5
  150 LEV(I)=FLEV
  160 CONTINUE
      IF (NX.GT.64) NX=64
      IF (NY.GT.64) NY=64
      DO 180 I=1,NX
      DO 180 J=1,NY
      KLT=1
      DO 170 K=1,32
```

```
      IF (IA(I,J).GE.LEV(K)) KLT=K
  170 CONTINUE
      IB(I,J)=KLT
  180 CONTINUE
C
C
C     ONCE IB HAS BEEN COMPUTED, THE PICTURE CAN
C     BE PRINTED.  EACH POINT IN THE PICTURE CAN
C     CONSIST OF UP TO FIVE CHARACTERS OVERPRINTED
C     ON ONE ANOTHER.  SINCE THERE ARE 32 POSSIBLE
C     GRAY LEVELS, THERE IS A 32X5 MATRIX, GRAY,
C     THAT CONTAINS ALL OF THE COMBINATIONS.  SINCE
C     EACH ELEMENT OF IB IS AN INTEGER BETWEEN
C     1 AND 32, IT CAN BE USED AS AN INDEX ON
C     GRAY TO OBTAIN THE CORRECT COMBINATION.
C     THE OUTPUT BUFFER, LINE, IS A 128X5 MATRIX.
C     EACH POINT IS OUTPUT TWICE HORIZONTALLY AND
C     ONCE VERTICALLY TO ATTEMPT TO COMPENSATE
C     FOR THE SPACING OF THE PRINTER.  HENCE, THE
C     FULL OUTPUT BUFFER REPRESENTS ONE ROW OF IB.
C     AS THE ROW IS GENERATED THERE IS A VECTOR,
C     BLANK, THAT INDICATES WHETHER OR NOT ANY
C     NON-BLANK CHARACTERS ARE PRESENT IN THE
C     BUFFER.  IF THERE ARE NOT ANY, THAT ROW IS
C     NOT PRINTED.  THIS SPEEDS UP THE PRINTING
C     PROCESS.
C
      WRITE(LG,1)
      IX=NX
      IY=2*NY
      DO 210 I=1,IX
      DO 190 J=1,5
      BLANK(J)=0
  190 CONTINUE
      DO 200 K=2,IY,2
      J=K/2
      NG=IB(I,J)
      IF (NEG.EQ.0) NG=33-NG
      DO 200 L=1,5
      LINE(K-1,L)=GRAY(NG,L)
      LINE(K,L)=GRAY(NG,L)
      IF (NG.NE.32) BLANK(L)=1
  200 CONTINUE
      WRITE (LG,2)
      DO 210 L=1,5
      IF (BLANK(L).EQ.0) GO TO 210
      WRITE (LG,3) (LINE(M,L), M=1,IY)
  210 CONTINUE
    1 FORMAT (1H1)
    2 FORMAT (1H )
    3 FORMAT (1H+,3X,128A1)
      RETURN
      END
```

TAB I

IN THE UNITED STATES DISTRICT COURT

FOR THE DISTRICT OF DELAWARE,

POLAROID CORPORATION,	)	
Plaintiff,	)	
vs.	)	No. 06-738 (SLR)
HEWLETT-PACKARD,	)	
Defendant.	)	

** CONFIDENTIAL **

The deposition of Dr. Rangaraj M. Rangayyan, called for examination, taken pursuant to the provisions of the Code of Civil Procedure and Rules of the Supreme Court of the State of Illinois pertaining to the taking of depositions for the purpose of discovery taken before LISA SCHWAM, CSR No. 84-004650, a Notary Public within and for the County of Cook, State of Illinois, and a Certified Shorthand Reporter of said state, at 54th Floor, 200 East Randolph Street, Chicago, Illinois, commencing, on the 9th day of May, A.D. 2008, at 8:07 a.m.

the subroutine DSP as a subroutine actually used to display images on a line printer?

A. That's correct.

13:08:58 Q. The first subroutine in the Gonzalez algorithm listed on page 451 is called CONVRT(AR)?

A. Yes.

Q. Do you see that?

This subroutine converts character data into

13:09:28 integer data; isn't that right?

A. That is what it states there, yes.

Q. The subroutine CONVRT(AR) that I just described specifically converts character data ranging from zero to nine and A to V; is that right?

A. Yes.

13:10:01 Q. The character data ranging from zero through nine and A through V described in subroutine CONVRT(AR) represents 32 gray levels, don't they?

A. Correct.

13:10:27 Q. The subroutine CONVRT(AR) converts the character data into integer data ranging from zero to 31, right?

A. Correct.

Q. The integer data that ranges from zero to 31

13:10:58 represents 32 gray levels; is that right?

A. In this context, yes. It's a way of coding 32 values using only one character per value.

Q. So the dynamic range for images that are converted, that have pixel data converted to the integer

13:11:28 data, is zero to 31, right?

A. Within the context of that subroutine, yes. There are other meanings outside.

Q. What do you mean? There are other meanings outside what?

A. The understanding, this particular way of

13:11:55 representation of images, was one way that Gonzalez and Wintz could provide image data of the kind you'll see later in this same appendix in the book. I'm sorry, not the same appendix, but Appendix B.

In Gonzalez and Wintz, there are examples of images. There are encoded representations using this

13:12:29 code. And this code is limited to 32 shades of gray. An image provided in this code is limited to, as you said, 32 levels of gray.

Q. So in the particular application of the Gonzalez algorithm, the dynamic range is zero to 31; is that right?

13:12:58 A. I need to confirm that. I have used this before many years ago. There are two parts to this.

So I would not characterize the '381 patent's method as global, but it uses local as well as global measures derived from the image.

Q. And so with respect to the Gonzalez algorithm, what do you contend that it demonstrates that it is using a local characteristic of the image to determine an output pixel value as you just defined the meaning of "local"?

A. If the locality is defined as the entire image, yes, it does use measures that are derived from the image. And as I mentioned earlier, one could also modify the definitions of SS and FLEV to include other measures of the image.

Q. So in your opinion, when a contrast enhancement algorithm relies on global operators solely, do you consider that a local contrast enhancement mechanism?

A. If it uses solely, as you mentioned, global measures, no, it cannot be local. But I have shown in my supplemental report that applying certain modifications as expressed in the Polaroid's expert report, it is possible to modify these definitions to include local variables and the ratio Av over M as in my supplemental report.

Q. Prior to performing the modifications that you

describe in your supplemental expert report, isn't it true that the Gonzalez algorithm is a global transformation scheme?

A. If you take it on the face of what is present on page 454 or in this Exhibit 682, I agree with you. But I have mentioned that one who knows image processing could easily modify that method to include local variables.

Q. I could see you're just itching to tell me all about it so let's ...

A. I'm sorry. I didn't get that.

Q. I said I can see you're just itching to tell me all about it so let's go to Exhibit 672, which is your supplemental expert report in this case.

A. May I go back and ask you to characterize what you meant by what you said.

Q. When?

A. Just now.

Q. I'm not sure what you're referring to.

THE WITNESS: Could you read the previous statement made.

MR. PEREZ-ALBUERNA: Hold on. Let's take a break. Let's go off the record.

THE VIDEOGRAPHER: Going off the video

TAB J

Most computer graphics used in the communications areas is two-dimensional and is in the area of post-production, such as titles and special effects. What is needed most is an artist who can use a computer system. The basics of art, color theory and visual skills are essential. In addition, knowledge of a variety of applications is desirable, as well as some programming in order to understand the applications better.

As mentioned in the area of theatre, there is also increased use of computers in administrative departments in all areas. Hence, using graphical techniques to display information has led to graphics specialists in information centers within the data-processing departments of large companies. People with graphic skills who are interested in business computing may have opportunities to carve out interesting career niches creating informational displays.

We have summarized the skills relevant to each of a number of disciplines. Frequently, however, projects are accomplished by a team of specialists. While it may be beneficial for the artist to understand programming and for the programmer to possess artistic ability, it may not be essential. It is sometimes more efficient for them to work together to maximize their talents, which is why communication skills are very important. One of the most pertinent examples of this is in the area of computer-aided design and computer-aided instruction. Designing instructional software might involve a highly technical, efficient programmer, an instructional design specialist who understands the user interface and ergonomic aspects, a content specialist and a graphics artist.

ACM has formed a committee to study and report on computer careers. The SIGGRAPH careers group will continue to work closely with this committee to provide any information we can. We look forward to seeing the results of this study.

Additional information

1. *Computer Graphics Marketplace.* Oryx Press, ISBN 0-89774-0866.
   This publication lists manufacturers, consultants and services, professional organizations, educational programs, conferences and conventions and publications.
2. *Computer Graphics Directory,* PennWell Directories, a division of PennWell Publishing Company, Box 1260, Tulsa, Oklahoma 74101 (918-663-4225).
   This publication lists company profiles in addition to descriptions of hardware.
3. *The S. Klein Directory of Computer Graphics Suppliers: Hardware, Software, Systems and Services,* Technology and Business Communications, Inc., 730 Boston Post Road, Sudbury, Massachusetts 01776.
   This publication includes directories of consultants and educational sources in addition to vendor information.
4. "The Motion Makers," *AVVIDEO,* June, 1984.
   This is a list of visual service companies which specialize in creating animation and special effects.
5. *Computer Graphics Education Directory* by Steve Cunningham, *Computer Graphics,* October 1985.
   This describes college-level computer graphics courses in a variety of disciplines.
6. *The Computer Careers Handbook* by Connie Winkler, Arco Publishing, Inc., 215 Park Avenue South, New York, NY 10003.
   Most of the career concepts in this book also apply to computer graphics careers.
7. *A Survey of Computer Aided Drafting and Design in Kentucky and Tennessee,* Dr. Kenneth Mussnug, Department of Industrial and Engineering Technology, Western Kentucky University, Bowling Green, Kentucky 42101.
8. "Getting Inside Molecules" by Margaret Neal, *IEEE Computer Graphics and Applications,* October, 1985.
9. *Literature in Computer Graphics for the Year 1984: A Bibliography* by G.F. Schrack, published in May 1985 *Computer Graphics.*

A Tutorial On Developing A Computer-Controlled Camera System

Neil Sullivan
C. Durward Rogers
Stephen Daniel

Microelectronics Center of North Carolina
3021 Cornwallis Road
Research Triangle Park, NC 27709

Abstract

A research organization or graphics house often has a need for custom pictures for presentations or other purposes. These pictures may be prohibitively expensive, if not impossible to create, if they are produced by a commercial art firm. For groups that have such a need, an in-house computer-controlled camera may be a solution. To develop a quality system, users will have to overcome a series of electronic, mechanical, photographic and computer-related hurdles. This tutorial deals with adjustments to the system in four classes to achieve the best possible pictures. The adjustments are for clean signal, image placement and focus, contrast and brightness, and software interface. In each case, common problems are discussed, some options are given, and the chosen solution is explained.

CR Categories: *I.3.4 Graphics Utilities,* application packages, software support; *I.4.9 Applications.*

Other Keywords: gamma correction, rgb signal correction, color compensation, photography, film.

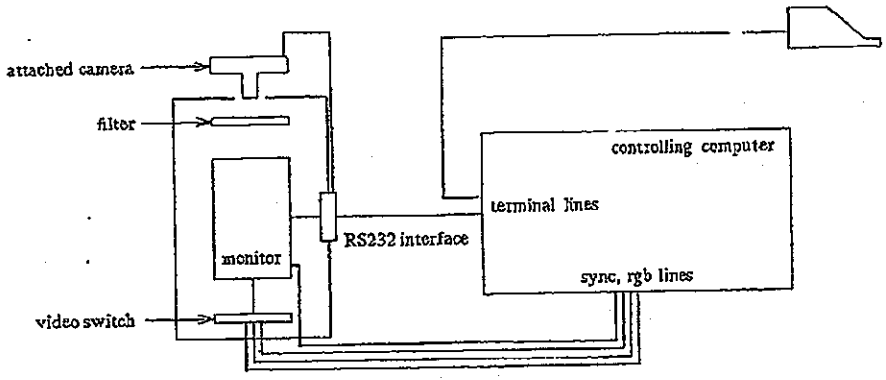
1. Overview

1.1. What the system is

Our system consists of a VAX[1] 11/780 computer running UNIX,[2] a Lexidata[3] 3400 high resolution (1024 X 1280) graphics device, and a Dunn Instruments 632 color camera. The VAX communicates with the Dunn camera through its RS232 microprocessor interface, and the Lexidata generates an rgb signal to a Tektronix[4] 634 monitor, which is housed in the Dunn. The monitor is black-and-white, flat screened, high resolution, and the image is controlled by a video switch. Various attachable cameras, or recorders, including 35mm still, 16mm cine,

Polaroid 4X5, and Polaroid 8X10 cameras are available. The cameras attach to the control box via a mounting assembly on the box, and they view the screen through colored filters. The system is diagramed in figure 1. Note that a variety of companies manufacture similar equipment, to which this tutorial is also pertinent.

owning organization to get the best possible performance from the system, including the best quality pictures and a simple software interface. In trying to reach such a goal we encountered some common problems: achieving a clean signal, image placement and focusing, brightness and contrast, and the design of the software interface.



System Diagram
Figure 1

1.2. How the system works

The RS232 microprocessor interface operates the attached camera and the video switch, sets the filter position, and controls the image on the monitor. The lens of the attached camera is exposed to the black-and-white monitor screen through a colored or clear filter. When the microprocessor receives a normal expose command, the shutter of the attached camera is opened; the video switch sequences through the red, green, and blue signals; and, depending upon the signal currently on the monitor, the appropriate filter is moved before the lens. If the picture is to be black-and-white, the clear filter is placed before the lens, and the video switch exposes the green signal to the attached camera.

1.3. System uses and common problems

The system is extremely useful for research organizations or graphics houses that need to make custom pictures that cannot be easily produced by a commercial art firm. For a graphics house, the usefulness is obvious, but it is perhaps not as obvious how a research organization could significantly benefit. In our case, we are frequently required to present VLSI design research developments to professional and lay audiences. The production of slides relating to our research is extraordinarily expensive when done by outside graphics firms. Whatever the case, it is to the benefit of the camera-

2. Clean signal

Cable, sync, and signal overlap are three major causes of a poor signal, and a bad signal most often results in ghosting on the image. The cables used for signal propagation in our case were standard 75 ohm coaxial cables. Cables must have a good shielding to protect against stray electromagnetic interference and to ensure a constant impedance.

Cables must be terminated well. Coaxial cable is notorious for losses through connections and poor terminations. All termination points are sources of loss, so it is imperative that one and only one termination be used per line. A good termination or connection must have an impedance equal to that of the line. All cables and terminations must be 75 ohm, not 50 ohm. All signals must follow a single path and be terminated at the far end only. Tee-connections should be avoided, but if they are absolutely necessary, the arm of the tee must be as short as possible, and its ends must be unterminated since the main line of the cable will be terminated. Note that some equipment has internal terminators in place at all times, some has loop-through connectors, and some has switches that allow you to select termination or not.

Another cable-related problem lies in determining proper length. Cables should be matched in length for each of the red, green, blue, and sync signals. Cables should also be as short as possible. If cables are of differing lengths, not correctly driven, or not properly terminated, ghosting rears its ugly head.

Ghosting can also be caused by the way sync is fed to the monitor. If the sync is bad, it may result from one of the previously mentioned problems: poor connections or mismatching coax length. Syncs can come into the system on a separate line, or it may be tapped off one of the rgb signals. Separate sync is usually 4V in amplitude, while sync combined with video is less than 300mV, and

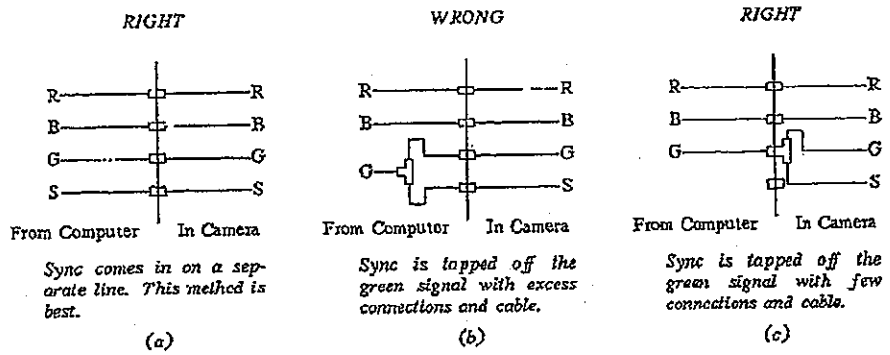
[1]VAX is a registered trademark of Digital Equipment Corporation.
[2]UNIX is a registered trademark of AT&T Bell Laboratories.
[3]Lexidata is a registered trademark of Lexidata Corporation.
[4]Tektronix is a registered trademark of Tektronix, Inc.

equipment that wants one form may not work properly with the other. In either case, if the sync is brought in poorly, ghosting is assured. Figure 2 shows three possible ways to tap sync. In figure 2a sync comes in on a separate line. In figures 2b and 2c sync is tapped off the green signal, but figure 2c shows the correct method, given the termination cautions listed above, because a minimum of connections and cable is used.

3. Image placement and focus

3.1. Image placement

Depending upon the monitor used, there may be electronic image placement adjustments. Image size may also be adjustable. There are a few problems involved with the placement and size of the image, relating primarily to image brightness and fitting the picture into the field of the attached camera's lens.

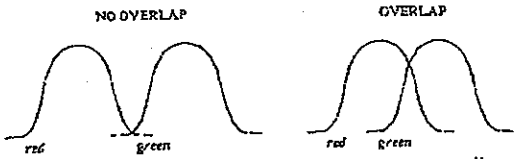


Three Ways to Tap Sync
Figure 2

Even after correcting obvious cable problems, the signal may still be bad. The signal can be checked more thoroughly by connecting an oscilloscope to the system to observe wave patterns. An excellent introduction to the subject, [ENNE], explains many of the concepts and requirements involved.

A basic problem that can be checked with an oscilloscope is signal overlap. Note that even though our monitor displays black-and-white images, these images are of red, green, and blue signals — all of which come into the monitor simultaneously. Only one is selected for the CRT at a time. If, on the scan of one line on the monitor, one color cuts in too early or one color does not cut off in time, there will be an overlap of colors that will appear like ghosting or blooming on pictures.

To test for overlap, full-screen-height images of red, green, and blue are needed on the monitor side-by-side. The oscilloscope traces are then connected to two of the adjacent signals, say red and green. The red signal should cut off before the green signal begins. If the previous signal does not cut off, then there is overlap, and the green portion of the image will be offset relative to the red portion. It is up to the manufacturer of the graphics device to repair this problem. See figure 3.

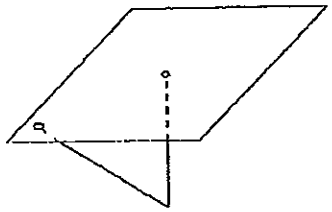


The curves represent the display of colors red and green on one scan line on the monitor. The left side shows red cutting off completely before the green signal is displayed. The right side shows red and green being displayed simultaneously for a portion of the scan line.

Signal Overlap
Figure 3

Obviously, the image size should be large enough so that it covers the entire field of view of the attached camera. This criterion is not always easily met. If the camera has vertical range of motion or if various attachable cameras have different fields of view, a median image size may have to be accepted.

The image should probably be placed as close to the center of the screen as possible. One reason for this placement is "pixel non-linearity." Pixel non-linearity basically means that the image is brightest on the screen where the electron beam is perpendicular, and that the brightness decreases toward the edges. If you are particularly worried about precise symmetric brightness of your pictures, it is to your advantage to keep the image in the center of the screen. Also, due partly to distortion, the picture quality is generally better in the middle of the screen. Both focus and resolution deteriorate as the image approaches the screen boundaries. See figure 4.



Pixel size changes towards the edges of the screen due to the varying angle and the greater distance from the electron gun. The result is a change in image brightness and resolution.

Pixel Non-Linearity
Figure 4

In our case, it was not possible to keep the image in the center of the screen, maintain proper image size, and have the image in the field of view of the lens. To solve the problem, we had the mounting assembly for the attached camera machined to enable a greater horizontal range of motion. The procedure was fairly inexpensive, and its major advantage was that the image did not have to be electronically moved every time we changed cameras.

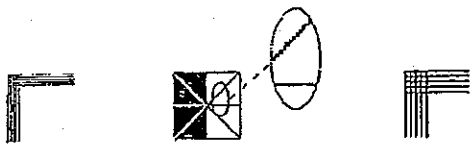
There are other adjustments to the monitor that should be checked. These adjustments will vary from monitor to monitor, depending upon the sophistication of the system. More advanced systems may have adjustments, such as orthogonality and edge-to-corner focus.

3.2. Focus patterns and correctable problems

The camera system has two subsystems that must be focused: the monitor and the attached camera. For focusing purposes, an intricate test pattern for the monitor is useful. The image should have qualities that enable checking not only for focus, but also for pixel blooming and ghosting.

The creation of such a pattern is much easier than it may appear. The major requirement is simply to have line patterns in different areas of the screen. To see how much resolution the monitor can handle, patterns of lines one-pixel-wide separated by one pixel, two-pixels-wide separated by two pixels, and so on work very well.

These lines are useful for checking all three problem areas: focus, blooming, and ghosting. Other figures should include lines that do not run strictly horizontal or vertical and solid areas of color. The diagonal lines are used for observing pixel definition. When a "beading" (or rastering) effect is seen on forty-five degree lines, the pixels are well defined. The blocks of solid color are particularly useful for blooming checks. It is important to note that all focus checks should include not just white patterns on a black background, but also black patterns on a white background. This double check ensures a more accurate assessment. See figure 5.



The end patterns are one-pixel wide lines separated by one pixel and two-pixel wide lines separated by two pixels. The center pattern uses 45 degree lines and a filled box to check for definition and blooming. The exploded view of the diagonal line demonstrates the beaded appearance characteristic of good pixel definition. It is recommended that other patterns and sizes be used in addition to these.

Possible Focus Patterns
Figure 5

Each of the three major problems can cause symptoms that resemble any of the other problems. For example, if there is ghosting or poor focus, colors will overlap, creating a blooming effect.

Problems can also be created by smoothing. Since visible scan lines can create incredibly ugly pictures, many systems smooth the lines away by moving each scan line from its home position up and down the width of one half a pixel. Trouble arises when the scan lines move more than they should, and pixels wind up overlapping. This activity is not well measured by the eye on the monitor; however, it is quite noticeable on pictures. A smoothing problem is generally characterized by vertical blooming. That is, scan lines will bleed into one another vertically, creating unwanted horizontal lines.

Another problem area is pixel size. If the image on the monitor is too bright, it will overdrive the phosphor and pixels will run into one another. The result is blooming. This situation is sometimes difficult to notice on a black-and-white monitor, but it can be corrected by taking pictures at various brightness and contrast positions to find the best possible setting.

3.3. Monitor focus

The monitor is more difficult to focus than the attached camera simply because of the eye strain incurred viewing the monitor. The eye strain is made a little worse since a magnifier of some type should be used to be able to really see pixels on the monitor. For our purposes, a 20X eyeglass lens was purchased from an optician. The 20X magnification was adequate, but 30X would have been more acceptable. There are 30X hand-held illuminated Micronta microscopes available at Radio Shack that seem to be ideal for this purpose. We purchased one, but the range of focus was not wide enough. We are currently modifying the mechanism.

The primary technique we used for focusing the monitor was to view the screen through the hand magnifier and adjust the focus until we could see the beads on the 45 degree line. It is important that uniform repeatable step sizes be used when focusing. That is, the adjustment knob should be marked so that reproducible changes can be made. If more than one person is adjusting the focus, the markings are used to identify the positions where each person finds the best focus.

After the general focus is adjusted, sophisticated systems may have other focusing adjustments that should be checked. For example, there may be an edge-to-corner focus. It is because of these extra monitor options and physical qualities of the phosphor and monitor that the focus pattern should cover the entire screen.

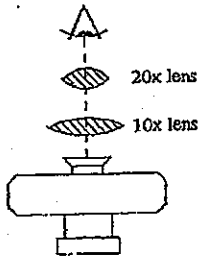
3.4. Lens focus

To focus the mechanical (lens) subsystem on the camera system, a more commonly photographic approach can be taken. Displaying the focus pattern on the monitor, the screen can be viewed through the eyepiece. The lens should be adjusted in uniform changes so less guesswork is involved finding the best position. It may be necessary to mark the lens barrel to ensure that repeatable changes are made to the focus.

The major problem we had was that the lens could not be reached to adjust it while it was mounted on the system. The procedure was then more time consuming, but not inherently different, simply because the attached camera had to be removed between each focus adjustment.

3.5. Precision focusing

We felt that a simple viewing of the screen through the attached camera was not a precise enough measurement of the best focus setting for either the monitor or the lens. For this reason, we took two more steps to complete the procedure. When viewing the screen through the camera lens, we "built" a "telescope" by using a 10X book magnifier and the 20X eyeglass lens. Laying the 10X magnifier over the camera eyepiece, we viewed the screen by looking through the 20X lens. The magnification enabled a much more precise focus setting. See figure 6. Depending on the manufacturer, it may be possible to purchase a focusing aid that attaches to the eyepiece of the camera.



Telescope For Enhanced Eye Focusing
Figure 6

Once both systems have been focused using the enhanced naked eye, we shot a roll of film of the focus pattern, adjusting the (mechanical or electronic) focus by a small constant amount between each picture. There are three important points to note about this test. First, we adjusted the camera (mechanical) focus using this test before precisely focusing the CRT. This order was chosen because it is much easier to reach an accurate setting of the electronic focus by eye. After the monitor is reasonably focused by eye, focusing the lens by this test first allows a much more exact setting of both the lens and then the monitor.

Second, it is imperative that the focus be changed by a uniform amount in this test. If the settings are not changed uniformly, it will be impossible to reset the focus to the proper place when the test is complete. And last, we recommend that only one focus system be adjusted for each roll of film to avoid confusion and to allow a wider range of precision in adjustments. The greater precision will enable a closer bracketing of the most acceptable focus.

When the film is developed, it should be viewed under a 10X-50X microscope. The 30X hand microscope with a light table worked very well. (Large magnifications with a slide projector should be avoided because of the generally poor quality of their lenses.) This viewing will allow the best possible setting to be selected. This test was indeed time-consuming, but the superiority of the post-test slides was worth the time involved.

4. Brightness and contrast

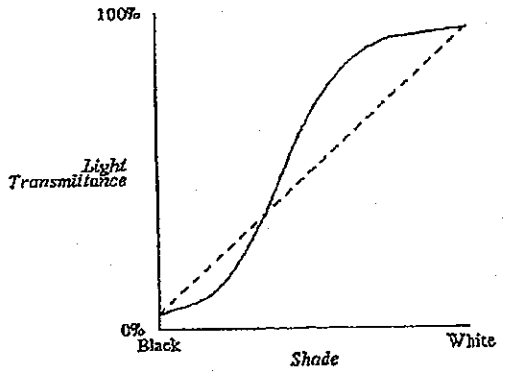
4.1. Gamma curves

When pictures are taken, it would be nice if what appeared as black, 80% black, or 30% black on the monitor appeared as exactly that on the film. Such is not the case. Some background information is required to explain this situation.

Film has a reaction to light that is not linear. To a serious photographer this reaction is a problem. The reaction creates an S-shaped curve called a gamma curve, D-log E curve, or characteristic curve. The curve represents the range of light intensities that the film can handle. For a photographer of computer images, the problem is two-fold: the phosphor in the monitor has a similar property. The brightness of a pixel is not a linear function of the signal that displays the pixel.

It is important to ensure linear response because many algorithms assume it. When lines, other than those strictly horizontal or vertical, are drawn on a graphics screen, they must appear jagged because of the limited resolution of the pixels. This jagged effect is sometimes corrected by anti-aliasing algorithms that assume a linear response. Non-linearity interferes with the correction and degrades the image produced. Indeed, since the actual nature of the non-linearity (the shape of the gamma curve) varies from display to display, from film lot to film lot, and over time, we need to measure these non-linearities just to obtain consistent and repeatable results.

Figure 7 illustrates a typical gamma curve. The gamma curve measures light transmittance through film. The graph labels are shade (exposure) and percent of light transmittance (density). The curve describes the relationship between exposure, development, and film density. It shows that as we approach blackest black and whitest white, equivalent increases (decreases) in light to the film affect the film much less than in the center shades of grey. It is to those shades of grey in the middle that the film responds linearly.



Gamma Curve: Transmittance Versus Shade
Figure 7

In our case, the shade axis of the graph increments from black to white in values from 0 to 255. The shades, although given numerical values, are converted by a

digital to analog converter (DAC) to voltage values that are used to control the intensity on the monitor. Since our gamma curve must measure both the film and its subject, the curve of figure 7 represents the combined effect of the non-linearities in both the monitor and the film.

All gamma curves are not identical. The shape of the curve reveals a great deal about the film. As the curve moves left or right, the speed of the film changes. When it moves left, the film is faster, and less light is needed to expose it. As the slope of the curve increases, the film has greater contrast. The gamma curve of the phosphor will behave similarly.

Photographers have the general problem of matching the contrast range of the subject to the contrast range of the film. The range of one is usually greater than the range of the other. Here the matching is complicated by the gamma properties of the monitor. An ideal world would have monitors and films that matched easily. Unfortunately, we do not live in an ideal world. It is a good idea to have the contrast ranges as close as possible, and this can be done by changing films for a better match. For example, Polaroid 4X5 film has a very steep curve, yielding a sharp contrast, but Ektachrome 64 has a much slower rising curve, allowing for subtle shade changes. We could alternately shrink or expand the contrast range of the monitor (subject) to match the film, but since a variety of cameras and films will be used, we opted to try to avoid as many electronic adjustments as possible, such as the contrast range of the monitor. When subject and film contrast ranges match, the result is a full-scaled picture. If the film range is shorter than the subject range, pictures are over-scaled, and they look too contrasty. Pictures appear flat if they are under-scaled. See figure 8.

difficult path promises better results. If you cannot adjust contrast and brightness on your system, or even if you are willing to accept the manufacturer's recommended setting, you can simply try to calculate a linear curve from the gamma curve. A useful reference here is [COWA].

A densitometer, which is used for measuring light transmittance, will be needed. A roll of film is shot of all the shades of grey from white to black, and the amount of light transmitted through the picture is measured by the densitometer. Even though the densitometer performs the measuring operations, pictures are often viewed by eye to check for color tones and other aesthetic qualities. Careful observation of the pictures will help the user develop a critical judgment of computer images and an understanding of the process, which can save considerable time and effort when new films or systems are calibrated. For this reason, we recommend that positive slide film be used. If negative film is used, such viewing will be difficult because the colors are complemented, and the negatives are covered by a printing mask. Prints are not reliable because of developing and printing inconsistencies.

Depending on how many shades of grey the monitor can display, it is a good idea to include all shades on one test roll. However, all shades will probably not fit on a single roll of film. Here is where we run into our first calculation trade-off. There are three choices, each with peculiar disadvantages. We can take a picture of every shade, putting more than one shade per slide. We can take a picture of every shade, putting only one shade per slide and using multiple rolls of film. Or we can take a picture of every other shade or every fourth shade, using as few rolls of film as possible.

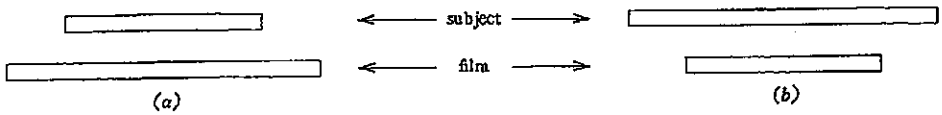


Figure (a) shows a subject contrast range less than the film contrast range. Resulting pictures will have little contrast and appear dull. Figure (b) shows a subject contrast range that is greater than the film range. Resulting pictures will be overly contrasty, causing loss of detail in extremely bright and dark areas on pictures.

Mismatching Contrast Ranges
Figure 8

Even if films are changed, it is quite impossible to align the gamma curves precisely. We can, however, do the next best thing: calculate the gamma curve of the developed pictures and attempt to linearize it. We shall assume that the curve of the monitor is unchangeable and therefore constant. By photographing the monitor, and calculating the curve, we can linearize that curve, resulting in pictures of good quality. An in-depth treatment of gamma curves can be found in [CATM] and [WHIT].

4.2. Problem division into two paths

4.2.1. Path one

At this point, the problem splits into two paths. One path is easier than the other, but as usual, the more

The big disadvantage to more than one shade per picture is inconsistent readings over the surface of the image. Two things can cause different readings on various parts of the picture. The first situation is non-uniform spreading of phosphor on the monitor. If the phosphor is not evenly spread, the electron beam will excite thinner areas more brightly than thicker areas. The second situation you run into, even if the phosphor is uniform, is dimmer pixels towards the edge of the screen, due to non-linearity (see figure 4).

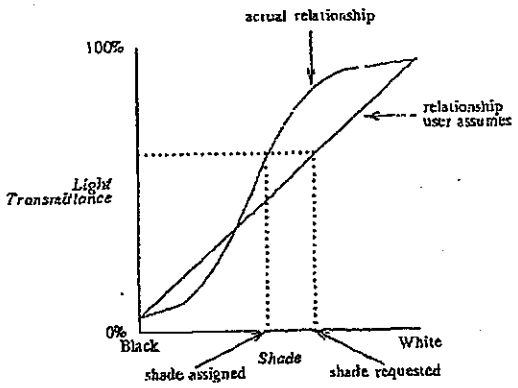
The major disadvantage to using multiple rolls is in the consistency of development. Developing and printing procedures have inherent inaccuracies that change the quality of pictures and the shape of the gamma curve from

roll to roll. Expecting precise calculations from multiple rolls of film is not always wise.

The problem with recording only certain shades is simply that not all shade values are exactly known. However, we chose this third method since we had 256 shades to measure. We used every fourth shade and minimized our film use to two rolls, without leaving too many unrecorded shades.

To avoid inconsistent readings between pictures, we took all measurements with the densitometer from a distinctly defined center square, and to minimize the problem of multiple rolls, we instructed our processing lab to develop our second roll of film immediately after the first, as if the two were one roll. To fill in the missing values between the shades, we used a cubic spline interpolation in which tangents are given. A cubic spline was chosen since the approximation has a smooth, predictable behavior and the point prediction is very accurate due to the shape of the typical gamma curve.

By recording the measurements and approximating the intermediate points, we obtained the gamma curve of the complete system, which is actually a combination of the curves for the phosphor, the film, and the development. Then, by finding the slope between the two endpoints of the curve, we found the equation of the line that connected them. We then wrote a program to compute the transformation between the curve and the desired line. The program calculated a table, which is accessed each time the user requests a shade. The shade requested is mapped to what the table considers the best choice, and that shade is displayed. A more efficient algorithm loads the table directly into the color map. See figure 9.



The Functional Operation Of A Linearized Gamma Curve
Figure 9

We created the table so that there were no duplicate shades, with increments as equal as possible on the line. We naturally lost a few shades from the ends of the curve, but the loss was not important since those shades were so closely grouped.

An example will more easily illustrate this process. Consider figure 10. Table 1 shows some of the values for the line from our actual data. Associated with each shade number is the percentage of light transmittance through the film. The percentages correspond to the percent of full white (and distance from full black). That is, shade 167 is 66.32% of white.

Shade	Percent
⋮	
166	66.02%
167	66.32%
168	66.56%
169	66.90%
170	67.23%
⋮	

Table representing the desired linear curve.

Table 1

Percent	Shade
⋮	
66.02%	110
66.32%	111
66.56%	112
66.90%	113
67.23%	114
⋮	

Table representing the actual gamma curve.

Table 2

Requested Shade	Assigned Shade
⋮	
166	110
167	111
168	112
169	113
170	114
⋮	

Table representing the functional relationship of the line and curve.

Table 3

The Functional Operation Of A Linearized Gamma Curve
In Table Form
Figure 10

Say we select shade 167 to display. We look into the table describing the desired intensity (table 1). Using the light transmittance percentage from the table and substituting that value into the equation for the curve as the light transmittance value, we can find the shade that should be displayed. Instead of an equation for the curve, we have another table representing its equation. So we read the corresponding shade number from table 2, given the light transmittance we found from table 1. So shade 111 is output. In practice this whole process is combined into one color compensation table, as shown in table 3.

4.2.2 Path two

Since we wanted to push the abilities of our monitor to the limit, to try to exact the greatest contrast range possible, we first had to determine what was the blackest black and the whitest white that could appear on the monitor. To start, we set the brightness to various settings and took pictures, until we found the shade of black where there was a noticeable difference between it and the film background black.

In practice, this result is difficult to achieve, if not impossible, but the test will yield some shade that can be considered the lowest usable output of the DAC. That is, the shade will be one more than zero — which is total black. Then we pushed the contrast to find the whitest white possible without causing blooming.

When looking for white we ensured that smoothing was on. If the feature is off, there will be dark areas between the scan lines that will cut down the overall brightness of the screen. Ideally, the white we want is just discernible from clear film base, but, again, reaching that state is improbable. Trying to reach pure white exceeded the ability of our monitor, but it was a good starting point.

With black and white found at this new brightness and contrast setting, we rechecked for blooming. Blooming was the limiting factor here — it destroys resolution. We can correct any blooming by an iterative process of reducing the contrast and checking the resolution. Then we simply proceed as in the first path and create the color compensation table.

4.3. Consistency

Consistency buys a great deal throughout this entire process, as does using quality materials. For our purposes, Ektachrome 64 Professional 35mm slide film was used. We used a low ASA to get sharp features. (Note: longer exposure time must be used.) If a "professional" film is used, colors are more consistent, but the film should be refrigerated. Instant slide or print films are useful for testing software or graphics packages, and Polaroid 4X5 or 8X10 prints are useful for image quality tests. However, because of the grain in current instant slides, they are not recommended for such tests.

To keep the film itself consistent, we chose to buy film in lots of twenty, with all rolls from the same emulsion batch. (It is also possible to buy film in 100-foot rolls for a bulk film loader.) All processing was done at the same professional lab, and we kept in-depth records describing all tests, their purposes, and their results. The records also contained information on all pictures taken, including exposure times, contrast and brightness settings, and any other unique picture qualities.

As we have stated before, developing is not always consistent. Rolls of film shot with identical pictures and developed one right after another will not necessarily have identical measurements on the densitometer. For this reason, if more than one roll of film is used to shoot the color scale, a numerical adjustment may have to be made to the readings of one or more rolls. When we used two rolls of film for the focus test, we made sure that the split between rolls was in the linear portion of the gamma curve so that a constant numerical offset could be used. We were not extremely concerned about the actual shape of the toe and shoulder of the curve — just about the end values of black and white. Given the loss of definition in the toe and shoulder, we believed that any developing inaccuracies were negligible.

System calibration is important. Calibration is a process whereby system characteristics are kept within a minimal range. Electronic and mechanical idiosyncrasies will cause characteristics to slip out of adjustment over the course of time — for example, absolute pixel brightness. Camera manufacturers describe a procedure to follow once or twice a year to reset the system values to ideal settings. If you have tailored the system to perform at particular values, such as a contrast and brightness setting to exact the most from the abilities of the monitor, you may have to adjust the calibration procedure accordingly. It is most important to keep the monitor image calibrated. An uncalibrated picture may become washed out, too bright, or lacking in contrast as the properties of the phosphors change.

Film can also be "calibrated." At a given contrast calibration, a type of film can be tested to find the best exposure setting. When that particular brand of film is used, those exposure times should be the default. We recommend that, if possible, only one type of film be used with the system because electronic adjustments may be necessary to achieve the best quality picture if path two of the gamma calculation is followed. The fewer adjustments to the camera that are made, the better.

4.4. Applying the gamma curve

There are some hints that may come in handy for calculating and applying the gamma curve. A densitometer requires a certain warm-up time, and we found that measurements drifted from time to time. So we always took four to five measurements, over the course of hours or days, and averaged the results. Our replication of measurements was well within an acceptable error margin.

Due to slight film and processing inconsistencies, the offsets that appear between multiple rolls of film are not constant. The shape of the curve is different, but the shape change is negligible since the greatest error is in the shoulder and toe of the curve. A constant numerical offset should suffice as a correction factor as long as the splits are made on the linear portion of the curve.

We could have performed the gamma test on all colors, not just black and white. Curves could be created for red, green, and blue, and those results could be applied separately just as these were. In fact, for the sake of curiosity, we did calculate those curves, and their shape matched the D log-E curves of the film manufacturer reasonably well. We felt, however, that the results from the black and white curve were sufficient for our purposes.

Also, placing a chosen frame, such as a full-screen white image, at the beginning of a roll of film will help the operator of an automatic mounting machine to synchronize frames and images. And if there is a mirror in the optical path, there will be a right/left or top/bottom reversal of the image. A print of the transparency will then be reversed. Such reversed slides are also, of course, reversed in a slide carousel and hence require refocusing when intermixed with conventional (unreversed) slides. It may be desirable to reverse images in the frame buffer before exposing a transparency to avoid these problems.

5. Software interface

5.1. General considerations

To avoid a cryptic interface to the camera, we designed a command language. Any language used to control a camera system will necessarily be a mixture of photographic and computer terminology. We chose to use as much photographic and as little computer terminology as possible since many non-technical people would be using the system. In some organizations, the reverse will be true. In either case, commands will certainly be needed to put an image on the monitor, set the exposure time, and expose the picture.

In some cases, it is not possible to use solely photographic terms and retain accuracy. For example, when setting exposure time, if the camera is set by sync pulses of the monitor and the user specifies the time in seconds, the conversion to sync pulses may not be precise. We were immediately faced, therefore, with an important decision. Should we tailor the language to the user or to precision? In such a circumstance, we took the philosophy that both ways of setting exposure time are acceptable, with a slight loss of accuracy if time is set by seconds. Inexperienced users can then set exposures in seconds and ignore the more complex commands, while more advanced users can obtain the precision they desire.

Exposure times should not require anything close to infinite precision, but there are some important factors to keep in mind. An exposure time set to an integral number of screen refreshes guarantees a uniformly shaded picture; although as exposure time increases, this point becomes less important because the number of screen refreshes may have grown so large that a fraction of a screen refresh will not create a noticeably inconsistent picture. It is very important, though, that exposures are made for at least the time of one screen refresh, or only a partial image will be photographed.

A major problem with any system will be synchronization. The camera system controller will be much slower than the controlling computer. It is up to the system designer to take this idiosyncrasy into account. The RS232 microprocessor interface will accept commands for the camera, and the camera system may or may not buffer them. Timing is most important for ensuring that no commands are lost and for ensuring that an image has been completely placed on the monitor before a picture is taken.

Synchronization and computer control are vital factors to consider when a camera system is to be purchased. Different systems will have different degrees of sophistication, and it is important to understand what the difficulties of synchronization will be and what control you will have over the camera before the system is bought.

If the computer can query the camera about readiness, using a type of rendezvous, the synchronization problem is solved. There may be a feature on the camera, as there was on ours, to send a ready-status signal to the computer. The computer had to check with the camera intermittently for readiness, and when the camera had finished the current command, it would send back an affirmative response.

The camera may have a command buffer. If there is no ready-status function, it will be the responsibility of the system designer to ensure that the buffer does not overflow, losing commands.

5.2. Language format choice

We were confronted with a choice of how to format the command language system: a file of command sequences or a subroutine library. In our case, we saw great advantages to each approach. The command file option allows the commands to be compiled so that parameter values could be checked for range and value correctness before execution. Compilation provides a great time- and film-saving service. Also, by using a command file format, the language could be used interactively by using the terminal as the command file. However, the subroutine library allowed the full power of programming language to be used, and loops and conditionals did not have to be re-invented for the command language.

We chose to use both options. Here software modularity was crucial. By placing procedures in a common library area, they could be called from either the controlling program that read a command file or from subroutine calls within a program, and consistency of the language system was maintained. In both formats, there were equivalent commands, and the photographic commands were identical in syntax.

Each format has the ability to interact with the operating system. Primarily, interaction enables images to be displayed on the camera system monitor, but during the course of exposing a roll of film, the user may want to manipulate operating system components, such as files. Operating system interaction should not be considered lightly. It is crucial to be able to check for the completion of procedures that display images on the monitor. To be able to make such a check from within the command language is mandatory.

The command file provides pre-execution error checks, and it has the ability to do single command, interactive testing. The subroutine library provides no pre-execution checks, but advanced programming structures are available. The former is best for instant work, novices, and 12 or 20 exposure 35mm rolls of film. The latter is best for advanced users, and it is a necessity for cine film production.

5.3. Debugging

Debugging is important when developing the software interface. Synchronization must be tested, and command sequences must be checked for proper operation. To help develop the software interface, an interactive language was useful, but we decided that command sequences for normal operation should be written in the form of files or programs because typing command sequences is a tedious and error-prone job. The interactive capability still exists for software and camera maintenance, and it is extremely useful for tests of graphics with a Polaroid camera.

A useful feature for each of the debugging areas is a trace file for each roll of film. Each run of the camera should log for each picture: picture number; name, if any; exposure time for each of the three primary colors; and total exposure time. If the picture is black-and-white, only the total exposure time should be given, and if the user gives commands to explicitly control the shutter, that situation should be noted. The trace file is an option that can be turned off in the event of movie film — where it would be far too long to serve any useful purpose.

The trace file proved irreplaceable for finding the best exposure settings for different films. It also came in handy when pictures had to be reshot, and we needed to know the best settings from previous rolls. Given the information we had about each picture, we could often check new problems with old test rolls of film. We also tried to put additional information into the pictures themselves, which was helpful when, inevitably, boxes of slides were jumbled up.

6. Summary

For our purposes, a quality camera system for in-house slide and film production was a must. A good software interface and gamma curve correction were our two main

objectives. We found that to reach our goals, we had to make adjustments to our system in four distinct areas. Adjustments to the signal helped most to correct ghosting. Changes to the image placement and focus improved alignment, resolution, and focus. Corrections to the brightness and contrast helped to fix blooming and linearize the brightness of pictures. Finally, an easy to use command language facilitated a general use of the system throughout our organization.

7. Acknowledgements

The authors are more than ordinarily indebted to Lars Nyland from the Microelectronics Center of North Carolina. They are merely ordinarily indebted to Jothy Rosenberg, Gary Nifong, Madison McCall, and Julia Slebos — all from MCNC. And they are less than ordinarily indebted to Ed Darken from Duke University, whose single contribution to this work was his suggested name for the command language (Askneil) — a suggestion at least one of the authors has yet to live down.

8. References

[CATM] Catmull, Edwin. "A Tutorial on Compensation Tables." *SIGGRAPH 79 Proceedings.* Association for Computing Machinery. 1979. pp. 1-7.

[COWA] Cowan, William. "An Inexpensive Scheme for Calibration of a Colour Monitor in Terms of CIE Standard Coordinates." *SIGGRAPH 83 Proceedings.* Association for Computing Machinery. 1983. pp. 315-321.

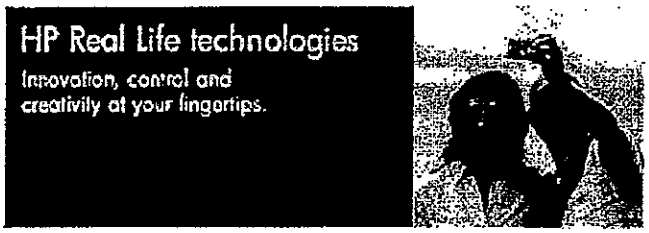
[ENNE] Ennes, Harold E. *Television Broadcasting: Equipment, Systems, and Operating Fundamentals.* Indianapolis, IN: Howard W. Sams & Company, Inc., 1976.

[SULL] Sullivan, Neil. *The Dunn Camera: A User's Guide to a Graphics Photographic Package.* Release 1.0. Microelectronics Center of North Carolina Technical Paper 85-19. Sep. 1985.

[WHIT] White, Minor; Zakia, Richard; and Lorenz, Peter. *The New Zone System Manual.* Dobbs Ferry, NY: Morgan Press, Inc., 1977.

TAB K

Real Life technologies™ in HP Photosmart printers and All-in-Ones

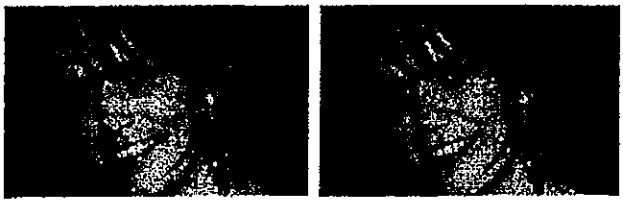


HP Smart Printing Features are designed to ensure you get the very most from your HP printer or HP All-in-One, helping you to achieve great results every time. HP-pioneered features benefit every stage of the printing process, from start to finish, and the intelligence that's built into HP printers or HP All-in-Ones puts real creativity at your fingertips.

» HP automatic red-eye removal
» HP Adaptive Lighting technology

HP automatic red-eye removal

Never see red-eyes on your photos again with your HP printer or your HP All-in-One. Remove red-eyes with HP automatic redeye removal. This unique, easy-to-use and highly effective feature automatically recognises red eyes on your photos produced by a flash.



Before **After**

» Back to top

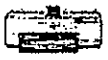
HP Adaptive Lighting technology

Get the light, right. This ground-breaking technology automatically adjusts high contrast photos to reveal the detail that would otherwise be lost in the shadows. Ideal for images with extreme light/shadow contrast or after using a flash.

Recommended Products


HP Photosmart A516 Compact Photo Printer


HP Photosmart A717 Compact Photo Printer


HP Photosmart 8253 Photo Printer (Q3470C)


HP Photosmart photo printers

Related links

» Real Life technologies™ in HP Photosmart digital cameras
» Real Life technologies™ in HP Photosmart printers and All-in-Ones
» Real Life technologies™ in HP Photosmart scanners

Before After

» Back to top

» Return to original page

TAB L
FULLY REDACTED

TAB M

US006813041B1

(12) **United States Patent**
Moroney et al.

(10) **Patent No.: US 6,813,041 B1**
(45) **Date of Patent: *Nov. 2, 2004**

(54) METHOD AND APPARATUS FOR PERFORMING LOCAL COLOR CORRECTION

(75) Inventors: Nathan Moroney, Mountain View, CA (US); Irwin Sobel, Menlo Park, CA (US)

(73) Assignee: Hewlett-Packard Development Company, L.P., Houston, TX (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: 09/540,162

(22) Filed: Mar. 31, 2000

(51) Int. Cl.[7] H04N 1/60; G06T 5/00; G06F 1/02

(52) U.S. Cl. 358/1.9; 358/518; 358/521; 358/519; 358/523; 382/263; 382/167; 708/277

(58) Field of Search 358/1.9, 518, 3.16, 358/517, 515, 3.06, 519, 521, 530, 532, 3.27, 523; 382/167, 162, 254, 260, 264; 708/277

(56) References Cited

U.S. PATENT DOCUMENTS

4,847,654	A *	7/1989	Homma et al.	358/300
5,189,529	A *	2/1993	Ishiwata et al.	358/451
5,282,036	A	1/1994	Worley, Jr. et al.	
5,793,855	A	8/1998	Peck	
6,028,957	A	2/2000	Hada et al.	
6,275,304	B1 *	8/2001	Eschbach et al.	358/1.9
2002/0186387	A1 *	12/2002	Moroney et al.	358/1.9

OTHER PUBLICATIONS

Moroney, "Local Color Correction Using Non-Linear Masking", IS&T SID 8th Color Imaging Conference, 2000.*
Myler et al., The Image Processing Algorithms in C, 1993, Prentice Hall P T R, pp 64–65; 82–84; 144; 220.*
Skarabot A. et al.: "Image sequence processing for videowall visualization" Nonlinear Image Processing XI, San Jose, Ca USA Jan. 24–25, 2000, vol. 3961, pp. 138–147, XP008016925.
Jain A K: "Fundamentals of Digital Image Processing" Fundamentals of Digital Image Processing Prentice-Hall International, Inc, US 1989, pp. 249–250, 264, XP002185234.
Chittineni C B: "Single Filters for Combined Image Geometric Manipulation and Enchancement" Proceedings of the SPIE, SPIE, Bellingham, Va, US, vol. 1903, Feb. 3, 1993, pp. 111–121, XP001028080.

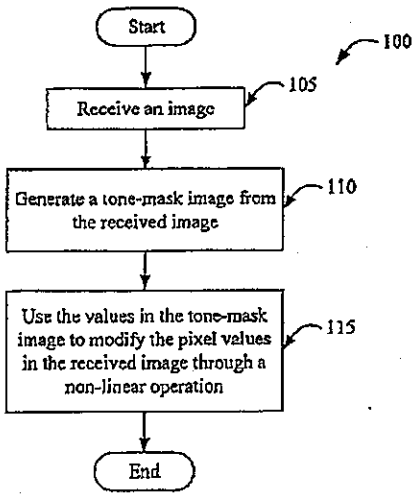
* cited by examiner

Primary Examiner—Scott Rogers

(57) ABSTRACT

The invention is directed towards method and apparatus for performing local color correction. One embodiment of the invention is a two-part process. The first part derives an image mask from an input image. In some embodiments, the mask is an inverted, low-pass filtered, monochrome version of the input image. The second part combines the derived mask with the input image through a non-linear operation. In some embodiments, the combination operation is a variable exponential function that has the mask values as part of its exponent and the pixel values as part of its base.

26 Claims, 7 Drawing Sheets



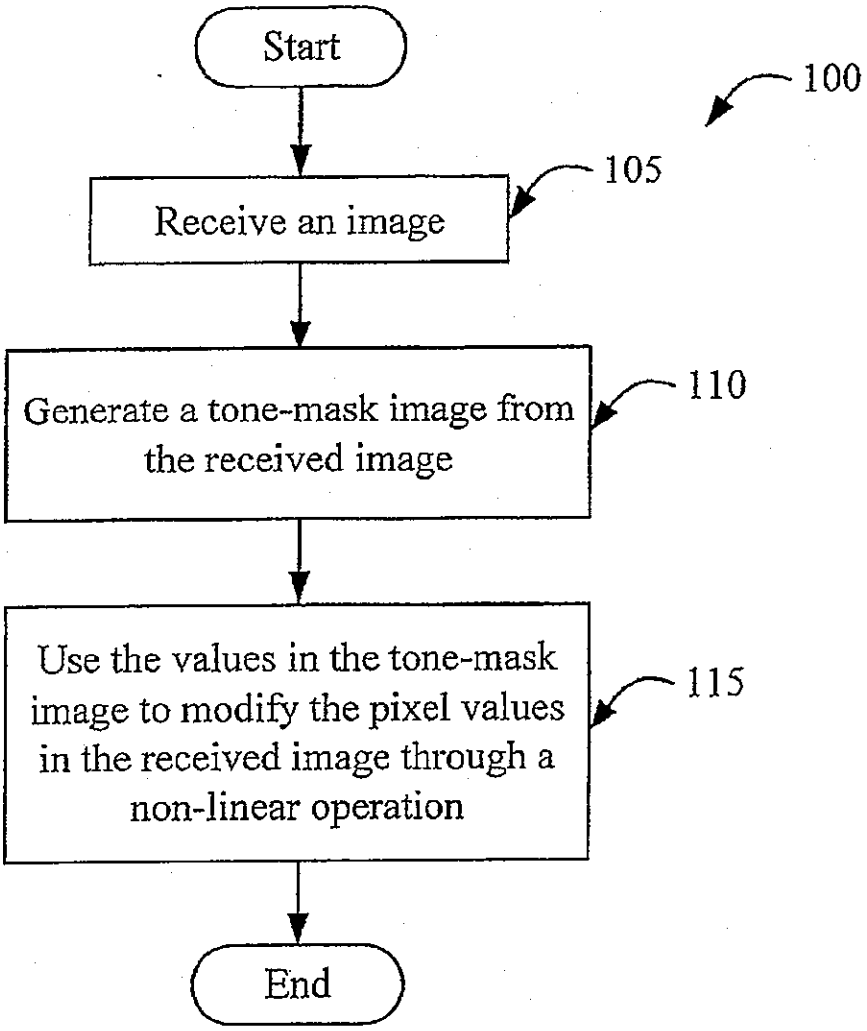

Start
100
105
Receive an image
110
Generate a tone-mask image from the received image
115
Use the values in the tone-mask image to modify the pixel values in the received image through a non-linear operation
End

Figure 1

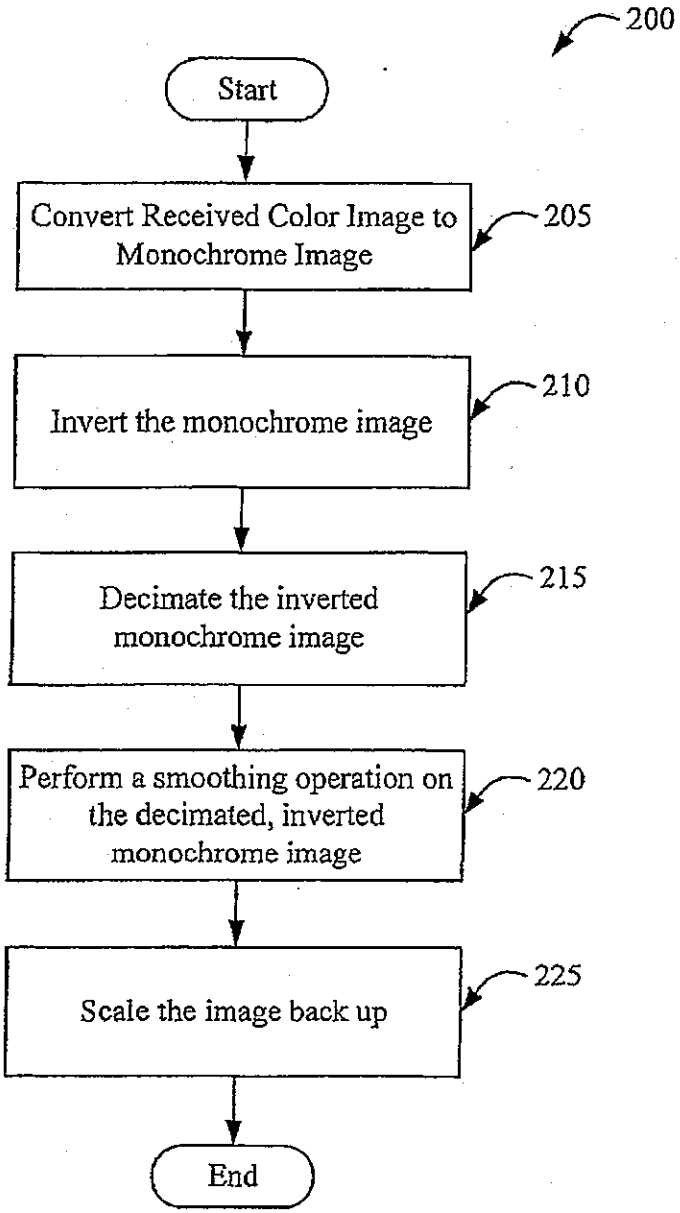

200
Start
Convert Received Color Image to Monochrome Image
205
Invert the monochrome image
210
Decimate the inverted monochrome image
215
Perform a smoothing operation on the decimated, inverted monochrome image
220
Scale the image back up
225
End

Figure 2

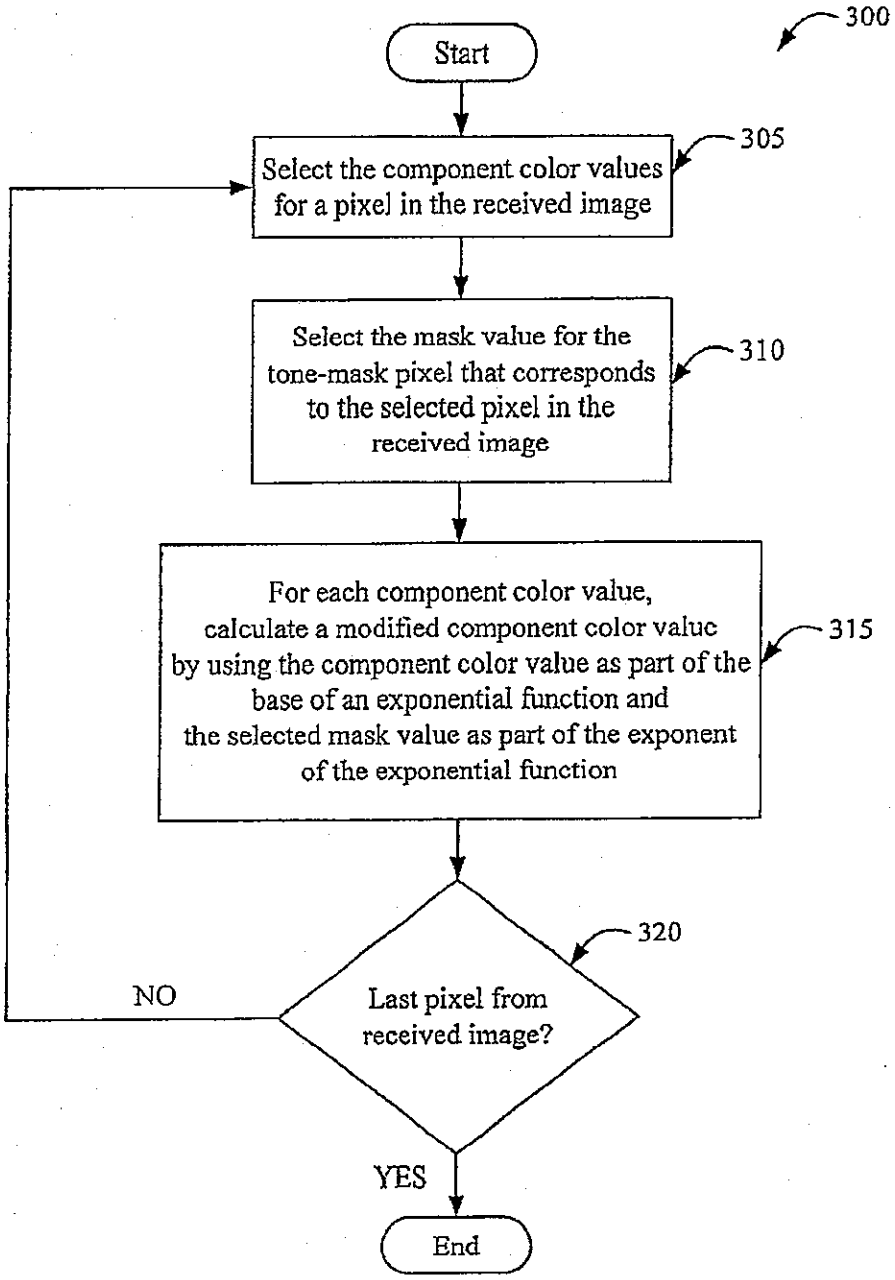

300
Start
Select the component color values for a pixel in the received image
305
Select the mask value for the tone-mask pixel that corresponds to the selected pixel in the received image
310
For each component color value, calculate a modified component color value by using the component color value as part of the base of an exponential function and the selected mask value as part of the exponent of the exponential function
315
Last pixel from received image?
320
NO
YES
End

Figure 3

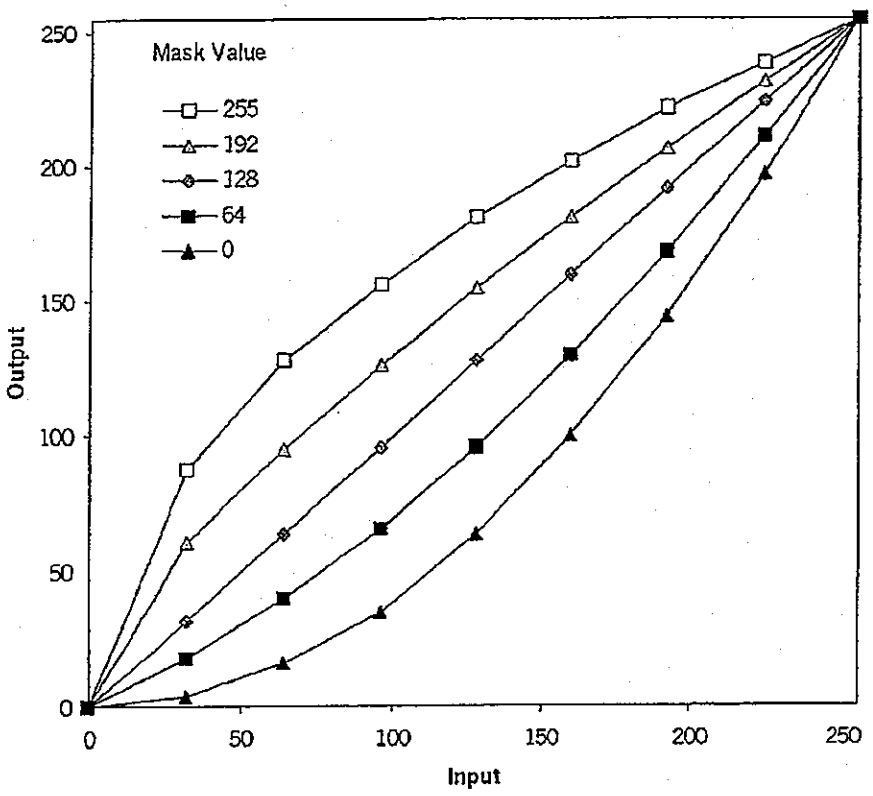


Figure 4

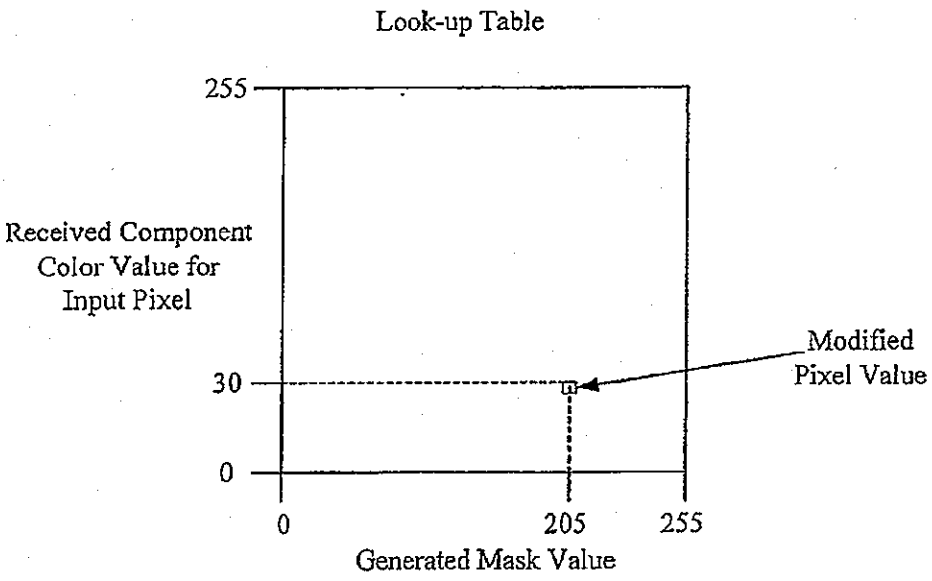

Look-up Table
255
Received Component Color Value for Input Pixel
30
0
Modified Pixel Value
0
205
255
Generated Mask Value

Figure 5

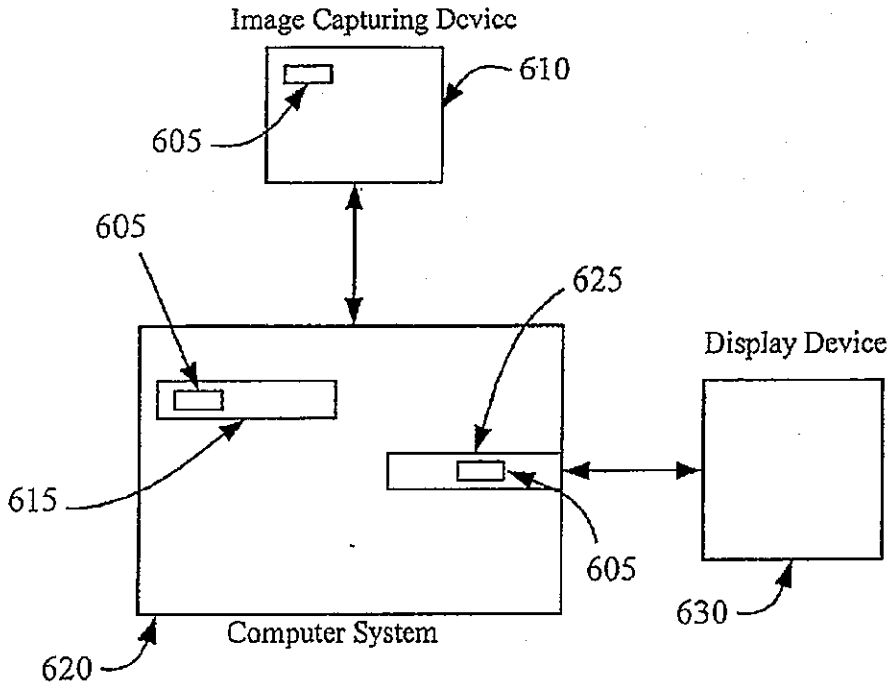


Image Capturing Device
610
605
605
625
Display Device
615
605
630
Computer System
620

Figure 6

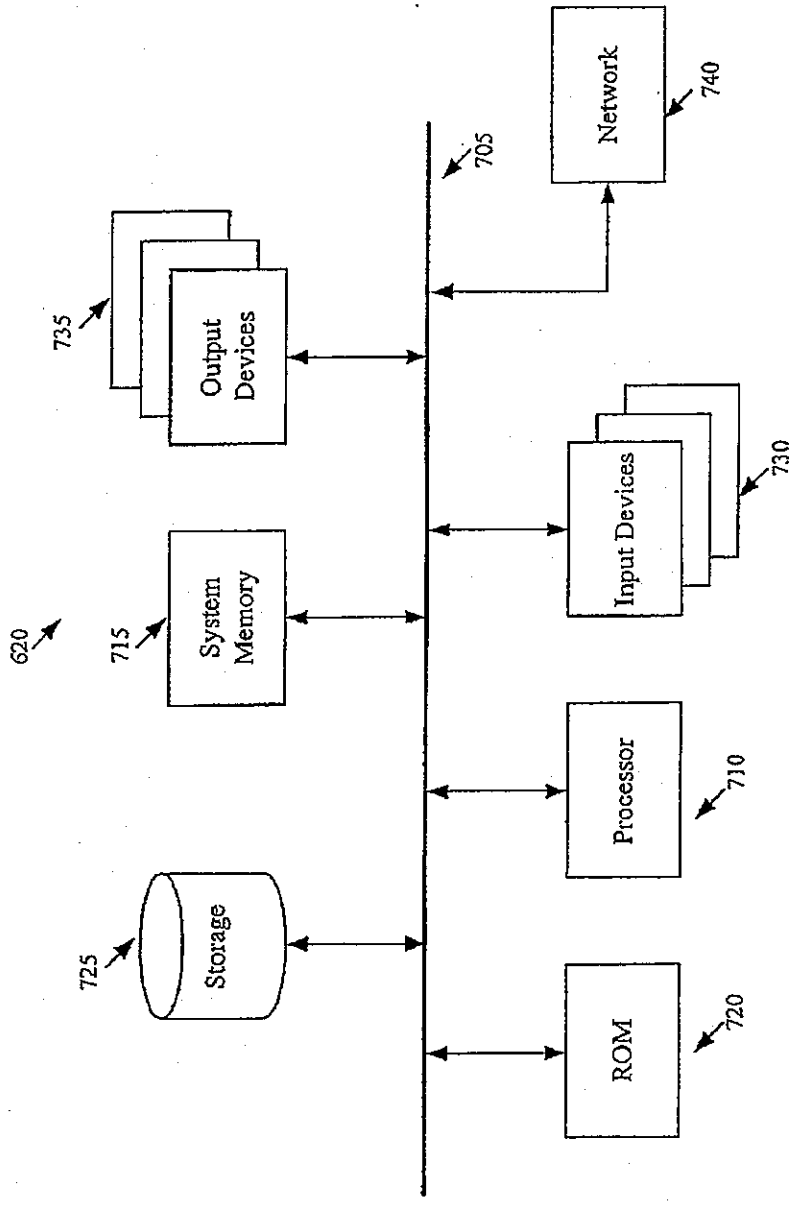

Network
740
705
Output Devices
735
Input Devices
730
620
System Memory
715
Processor
710
Storage
725
ROM
720

Figure 7

METHOD AND APPARATUS FOR PERFORMING LOCAL COLOR CORRECTION

The present invention is directed towards method and apparatus for performing local color correction.

BACKGROUND OF THE INVENTION

Color tone reproduction is an important aspect of image processing. Historically, tone correction has been performed on a global basis for a given image. Global tone-correction operations traditionally apply the same correction to all the pixels of an image. In other words, they are one-dimensional operations that map each input value to one and only one output value.

Global tone correction results in a reasonable correction when the dynamic range of the original image is fairly limited. When the original image has a large dynamic range, it becomes increasingly difficult to perform a global tone correction that will accommodate both shadow and highlight detail.

Consequently, several local tone-correction operations have been proposed for processing images with high dynamic range. A local tone-correction operation typically maps one input value to different output values, depending on the values of the neighboring pixels. This allows for simultaneous shadow and highlight adjustments.

Prior art local-tone correction techniques include manual operations, such as dodging and burning, and automated techniques, such as histogram equalization, piece-wise gamma correction, and Retinex algorithms. These techniques have shown considerable promise. Some of these techniques, however, are quite complex and time consuming. Others do not consistently provide acceptable quality. Therefore, there is a need in the art for a local tone correction technique that is simple and fast, and that produces high quality images.

SUMMARY OF THE INVENTION

The invention is directed towards method and apparatus for performing local color correction. One embodiment of the invention is a two-part process. The first part derives an image mask from an input image. In some embodiments, the mask is an inverted, low-pass filtered, monochrome version of the input image. The second part combines the derived mask with the input image through a non-linear operation. In some embodiments, the combination operation is a variable exponential function that has the mask values as part of its exponent and the pixel values as part of its base.

BRIEF DESCRIPTION OF THE DRAWINGS

The novel features of the invention are set forth in the appended claims. However, for purpose of explanation, several embodiments of the invention are set forth in the following figures.

FIG. 1 illustrates a local-color correction process that is used in some embodiments of the invention.

FIG. 2 illustrates a process for generating a tone mask through a low-pass filtering operation.

FIG. 3 illustrates a process for combining pixel values with their corresponding mask values from the tone-mask through a non-linear, exponential operation.

FIG. 4 illustrates several curves that show how pixel color values are mapped to output color values depending on the particular mask values.

FIG. 5 presents a conceptual illustration of a two-dimensional look-up table that stores modified pixel color values.

FIG. 6 illustrates several places in a computer and an image-capturing device for incorporating the invention.

FIG. 7 illustrates a computer system used to implement some embodiments of the invention.

DETAILED DESCRIPTION OF THE INVENTION

The invention is directed towards method and apparatus for performing local color correction. In the following description, numerous details are set forth for purpose of explanation. However, one of ordinary skill in the art will realize that the invention may be practiced without the use of these specific details. In other instances, well-known structures and devices are shown in block diagram form in order not to obscure the description of the invention with unnecessary detail.

FIG. 1 illustrates a process 100 that is used in some embodiments of the invention. As shown in this figure, the process 100 initially receives (at 105) a digital color image that is formed by a number of pixels (ie., picture elements). Specifically, in some embodiments, the process receives sets of component color values that represent the digital color image. Each set of the component color values specifies the color of a pixel of the digital image. In some embodiments of the invention, the component colors are the primary additive colors, red, green or blue ("RGB"). Other embodiments, however, use other component colors values (such as CMYK values, YCrCb values, CIELAB, CIECAM97s values, etc.) to specify the colors of the digital image's pixels.

As shown in FIG. 1, the process generates (at 110) a tone mask from the received digital image. A tone mask is an image that encodes the tone properties of the received digital image in an abbreviated form. In other words, the mask contains information about the color tone of the items in the received image, independent of the content or identity of the items. The numerical values in the tone mask are indicative of shadows, highlights, and midtones in the received digital image.

Like the received digital image, the tone-mask image includes a number of pixels. The color of each pixel in the tone-mask is specified by a mask color value. Each pixel in the tone-mask image corresponds to a pixel in the digital image. Similarly, the mask color value of each pixel in the tone mask is associated with the component color values of the corresponding pixel in the digital image.

Different embodiments of the invention use different techniques to generate the tone mask of the received digital image. Some embodiments generate this mask by performing an iterative operation on the received image. For instance, some embodiments generate the mask by using an iterative process (such as a Retinex process) that makes assumptions about the human visual system. Other embodiments generate this mask by performing a non-iterative operation on the received image. For example, some embodiments perform a low-pass filtering operation on the received image to generate this mask. One such approach is described below by reference to FIG. 2.

As shown in FIG. 1, after process 100 generates the tone-mask image, the process uses (at 115) the values in the tone mask to modify the pixel color values in the received image through a non-linear operation. The non-linear combination of the mask values with the pixel color values is in

contrast to previous techniques that combine the pixel and mask values through linear addition or subtraction operation. The non-linear application of the mask provides a gradual transition for the high contrast boundaries in the received image.

Different embodiments of the invention use different non-linear operations to modify the pixel color values with the mask values. Some embodiments use rotated, scaled sinusoidal functions, while others use sigmoidal functions. Yet other embodiments employ exponential functions that have the mask values as part of their exponent and the pixel values as part of their base. One such exponential operator is described further below by reference to FIG. 3.

FIG. 2 illustrates a process 200 for generating a tone mask through a low-pass filtering operation. As shown in this figure, the process initially converts (at 205) the received color image to a monochrome image (i.e., an image that only contains black and white pixel values, or contains black, white, and gray values). One way of generating a monochrome image would be to use the color values for one color component (e.g., for green) and discard the remaining color values. Another approach would be to compute a non-linear average of the component color values by using a scaling equation (such as 0.2R+0.7G+0.1B).

Next, the process inverts (at 210) the monochrome image. In other words, the process subtracts each pixel value from the maximum digital count, so that white values become black values, black values becomes white values, most gray values assume new gray values, and the mid-gray value remains the same. The process then decimates (at 215) the inverted monochrome image. Some embodiments decimate this image by selecting every n^{th} (e.g., 20^{th}) horizontal and vertical pixel in this image (i.e., selecting the pixel color value for every n^{th} pixel in the horizontal and vertical directions), and discarding the remaining pixels.

The process then performs (at 220) a smoothing operation on each pixel in the decimated, inverted, monochrome image. In some embodiment, this smoothing operation is a 3×3 convolution operation with a 1/9 kernel. In essence, such a convolution operation would replace each pixel value by the average value that is obtained by averaging a 3×3 pixel-neighborhood around the pixel.

Finally, the process scales (at 225) the smoothed, decimated, inverted, monochrome image back up to the resolution of the original received image. In some embodiments, the process performs this scaling operation by selecting each pixel value in the smoothed, decimated, inverted image as the pixel value for a n by n (e.g., 20 by 20) pixel area. The combination of the decimation, smoothing, and scaling operations provide the low-pass filtering operation of the process 200. The result is a tone mask in which there are no fine details and only large indistinct regions are visible.

The mask produced by process 200 is monochrome in order to avoid distorting the chroma of the image. This mask is inverted so that the power of the exponent for the combination operation is the opposite of the input value. For example, a light region in the received image will have a darker mask value (i.e., a higher mask value) and therefore will be darkened. Other embodiments of the invention do not invert the original image to generate the tone mask. Instead, these embodiments account for the need for inversion in the non-linear operator that is used to combine the mask with the received image.

In addition, the mask is decimated and then smoothed in order to speed up the process for generating the mask. Some embodiments of the invention, however, do not decimate the inverted monochrome image to generate a thumbnail image that can be quickly smoothed. Instead, these embodiments perform the smoothing operation directly on the inverted monochrome image.

The mask produced by process 200 is low-pass filtered (i.e., blurred) because otherwise the mask would flatten the contrast in the image. In other words, when the mask image is not blurred, then the non-linear combination of the mask and the received image will result in an image with a reduced image contrast. On the other hand, if the mask is overly blurred, then the invention's process reduces to simple gamma correction, and the local tone correction has less of an effect.

As discussed above, the invention uses the values in the tone-mask to modify the pixel values in the received image through a non-linear operation. FIG. 3 illustrates a process 300 for combining the pixel values with their corresponding mask values through a non-linear, exponential operation.

Process 300 initially retrieves (at 305) the component color values for a pixel in the received digital image. The process then retrieves (at 310) from the tone mask a mask value that corresponds to the selected component color values. Next, the process uses an exponential function to calculate (at 315) a modified component color value for each retrieved component color value. In some embodiments, the process uses a retrieved component color value as part of the base of the exponential function, and uses the retrieved mask value as part of exponent of this function.

Finally, the process determines (at 320) whether it has modified all the pixel color values in the received image. If not, the process repeats for the next pixel by returning to 305, and selecting the next component color value set for the next pixel. Otherwise, the process terminates.

Some embodiments of the invention use the following Equation A as the exponential function for modifying the component color values by their corresponding mask values.

$$O = 255 * \left(\frac{I}{255}\right)^{\left(2^{\left(\frac{C1 \cdot M - C2}{C2}\right)}\right)} \qquad (A)$$

In this equation, O represents the modified output color value, I represents an input component color value, M represents the mask value, and C1 and C2 represent constant values. Some embodiments of the invention use Equation A for pixel values of each component color. For instance, some embodiments use this equation on red, green, and blue values of each pixel in the received image, in order to obtain modified red, green, and blue values for all the image pixels.

The operation performed by Equation A is essentially akin to performing a pixel-wise gamma correction (i.e., a pixel by pixel gamma correction), where each pixel has its own particular gamma value (due to its particular mask value) that is determined by the tonal properties of the pixel and its surrounding pixels. Equation A assumes an 8-bit image and therefore uses a value of 255 to scale the data. However, higher or lower bit-depth images could be used, in which case the scale value would correspond to the maximum digital count.

Each mask value that is inserted in Equation A defines a tone-reproduction curve. In other words, for each mask value, a tone-reproduction curve can be used to represent how Equation A maps the input component color values to modified output color values. Similarly, a family of tone-reproduction curves can be used to illustrate this equation's

mapping of the input values to the modified output color values for all mask values.

FIG. 4 illustrates a family of tone-reproduction curves for the case where the constant C1 equal 1, constant C2 equals 128, and the mask values range from 0 to 255. In this figure, the x-axis is the input pixel value and the y-axis is the output value. The mask values are in the range 0 to 255 when an eight-bit color scheme is used (i.e., when each component color is represented by eight bits). In this example, mask values greater than 128 result in an exponent less than 1, while mask values less than 128 yield exponents greater than 1. Mask values equal to 128 result in exponents equal to 1 and does not change the input data.

The constants C1 and C2 determine the center and width of the family of tone-reproduction curves. Specifically, the constant C2 determines the center of the family of curves. In the example shown in FIG. 4, the center is the 45° line representing the mask value 128. The constant C1 determines the spreading of the curves in the family. Hence, changing the constant C2 moves the family of curves up or down, while changing the constant C1 spreads or contracts the family of curves.

Some interactive embodiments of the invention (1) allow their users to modify C1 and C2, and/or (2) automatically modify one or both of these constants based on the image properties. Modifying these constants changes the magnitude of the power function. Hence, such modifications change the degree of local tone correction performed by process 300 based on user preferences or image properties. For instance, if a received original image is very bright, a user can move up the family of tone-reproduction curves so that the process can darken the original image more in order to provide better contrast between the bright objects in the image.

Equation A can be implemented as a look up table, where the received pixel values and the generated mask values are used as indices into this table that identify the location of the pre-calculated modified pixel values for different combination of pixel and mask values. FIG. 5 presents a conceptual illustration of this approach.

The invention can be implemented at several stages in the image-processing pipeline. FIG. 6 illustrates several such stages. As shown in this figure, an image-capturing device 610 (e.g., a digital camera or a scanner) can use the invention's local-color-correction process 605 to improve the quality of the images captured by this device. The invention can be used to correct the color tone of images that were captured with uneven exposure or flash illumination. The invention can also be used to post-process images with a high dynamic range.

After capturing an image, the capturing device can quickly generate a tone-mask for the image by performing the fast and efficient process 200 of FIG. 2. The capturing device can then use a look-up table (such as the one conceptually illustrated by FIG. 5) to quickly identify the modified pixel values for each component color. In other words, the capturing device can use the original pixel values and the generated mask values as indices into the look-up table to retrieve modified pixel values.

As shown in FIG. 6, the invention's local-color-correction process 605 can also be part of an application 615 running on a computer system 620. The computer system can run this application to correct the color of an image captured by capturing device 610 or supplied by another application (such as a web browser application). The process 605 can also be implemented as part of a device driver 625 (such as a printer driver), so that the driver performs local-tone correction on an image before the display device 630 displays the image.

FIG. 7 presents a block diagram of a computer system 620 that is used in some embodiments of the invention. In some embodiments, the computer system 620 is separate from the image-capturing device 610. However, in other embodiments of the invention, this computer system 620 is part of the image-capturing device. Also, even though the computer system 620 is described below, one of ordinary skill will understand that other computer systems and architectures may be used in conjunction with the invention.

As shown in FIG. 7, the computer 620 includes a bus 705, a processor 710, a system memory 715, a read-only memory 720, a permanent storage device 725, input devices 730, and output devices 735. The bus 705 collectively represents all system, peripheral, and chipset buses that communicatively connect the numerous internal devices of the computer system 620. For instance, the bus 705 communicatively connects the processor 710 with the read-only memory 720, the system memory 715, and the permanent storage device 725. From these various memory units, the processor 710 retrieves instructions to execute and data to process.

The read-only-memory (ROM) 720 stores static data and instructions that are needed by the processor 710 and other modules of the computer system. The permanent storage device 725, on the other hand, is read-and-write memory device. This device is a non-volatile memory unit that stores instruction and data even when the computer 620 is off. Some embodiments of the invention use a mass-storage device (such as a magnetic or optical disk and its corresponding disk drive) as the permanent storage device 725. Other embodiments use a removable storage device (such as a floppy disk or zip® disk, and its corresponding disk drive) as the permanent storage device.

Like the permanent storage device 725, the system memory 715 is a read-and-write memory device. However, unlike storage device 725, the system memory is a volatile read-and-write memory, such as a random access memory. The system memory stores some of the instructions and data that the processor 710 needs at runtime. For instance, in some embodiments, the system memory 715 contains the look-up table that stores the modified pixel color values. Also, in some embodiments, the instructions necessary for carrying out the invention's process are stored in the system memory, the permanent storage device 725, and/or the read-only memory 720.

The bus 705 also connects to the input and output devices 730 and 735. The input devices enable the user to communicate information and select commands to the computer. The input devices 730 include alphanumeric keyboards and cursor-controllers. The output devices 735 display images generated by the computer. For instance, these devices display the modified digital images that the invention's local correction-techniques produce. The output devices include printers and display devices, such as cathode ray tubes (CRT) or liquid crystal displays (LCD).

Finally, as shown in FIG. 7, bus 705 also couples computer 620 to a network through a network adapter (not shown). In this manner, the computer can be a part of a network of computers (such as a local area network ("LAN"), a wide area network ("WAN"), or an Intranet) or a network of networks (such as the Internet).

The invention's local color-correction technique has numerous advantages. For instance, the invention's local correction can simultaneously lighten shadows and darken highlights, by using a simple pixel-wise "gamma" correction of the input data. Also, some embodiments balance global

and local contrast changes and reduce chroma distortions by using an inverted, low-pass filtered, monochrome version of the original image as the tone mask.

In addition, the invention provides a simple and fast method of computing and applying local tone correction. Its local color correction provides a significant improvement in image quality relative to a global correction. Its correction is also on par with the correction offered by other local tone correction methods, such Retinex or derivative algorithms. However, unlike other local tone correction methods, the invention does not need to rely on complicated visual models or masks. In sum, the invention provides a simple, computationally efficient, and easily parameterized framework for performing local color corrections.

The invention's local tone correction can be useful for enhancing or improving a several types of images. For example, conventional photography may yield an image in which one region of the image is underexposed while another region is correctly exposed. Flash photography is another instance in which one region in the image may be over-exposed while another is under-exposed. For these situations, a complex local correction technique based on the human visual system may be inappropriate. In addition, some image scenes have a wide dynamic range. The invention's simple and fast local correction process provides an ideal solution to improve quickly the image quality in these situations.

Lastly, the invention's local-tone correction is a complementary technology to traditional pixel based processing. For example, once the invention is used to correct the color of an image, the image can be reproduced on a given printer using a traditional pixel based pipeline. Use of a neighborhood image processing technique for one part of the imaging pipeline does not exclude subsequent pixel based image processing. This suggests that pixel and neighborhood imaging operations could be used where appropriate to optimize the overall quality and speed of the imaging pipeline.

One embodiment of the invention is implemented by using the Adobe PhotoShop application. Specifically, in this embodiment, this application can be used to (1) input an original image, (2) copy the original image into a second layer, (3) blur the image using the Gaussian blur filter with a radius of about 15, (4) invert the blurred image, (5) desaturate the blurred negative using the Hue/Saturation adjustment, and (6) use the Soft Light layer operator to combine the original and generated images to create the final image. Experimenting with the radius of the Photoshop blur filter shows how the overall image contrast is a function of the radius. In particular, the larger the radius the better the overall contrast, but if the radius is too large then the local tone correction has less of an effect.

While the invention has been described with reference to numerous specific details, one of ordinary skill in the art will recognize that the invention can be embodied in other specific forms without departing from the spirit of the invention. For instance, in the embodiments described above, one monochrome mask is used to modify pixel values for all component colors. One of ordinary skill, however, will understand that other embodiments use separate masks for each component color. In these embodiments, each of the masks can be optimized for its component color value. Thus, one of ordinary skill in the art would understand that the invention is not to be limited by the foregoing illustrative details, but rather is to be defined by the appended claims.

We claim:

1. A method of performing local color correction comprising:

(a) receiving an input image having a plurality of pixels, wherein each pixel of the input image is represented by a set of original pixel color values,

(b) generating a tone-mask image from the input image, said tone-mask image having a plurality of pixels each represented by a set of pixel mask values, wherein each set of original pixel color values has a corresponding set of pixel mask values, and

(c) determining a set of modified pixel color values for each set of original pixel color values, wherein each set of modified pixel color values equals a non-linear combination of a set of original pixel color values and its corresponding set of pixel mask values.

2. The method of claim 1 wherein determining a set of modified pixel color values includes using a set of original pixel color values and its corresponding set of pixel mask values as indices into a look-up table to retrieve the set of modified pixel color values.

3. The method of claim 1 wherein determining a set of modified pixel color values includes using a non-linear operation to combine a set of original pixel color values and its corresponding set of pixel mask values.

4. The method of claim 3 wherein the non-linear operation is not a subtraction or addition operation.

5. The method of claim 3 wherein the non-linear operation is an exponential operation having a base and an exponent, said operation using the set of original pixel color values as part of its base, and the corresponding set of pixel mask values as part of its exponent.

6. The method of claim 1 wherein generating the mask image includes using a non-iterative process.

7. The method of claim 1 wherein generating the tone-mask image includes performing a low-pass filtering operation on the input image.

8. The method of claim 7 wherein generating the tone-mask image further includes inverting the input image.

9. The method of claim 8 wherein generating the tone-mask image further includes converting the input image to a monochrome image.

10. The method of claim 1 wherein each set of original pixel color values is a set of component color values.

11. The method of claim 1 wherein each set of pixel mask values is a grayscale mask value.

12. A method of performing local color correction on an input image, said input image comprising a plurality of pixels, said method comprising:

(a) receiving a set of original color values for a pixel of the input image,

(b) generating a set of tone-mask values for the pixel of the input image, the set of tone mask values representing abbreviated color tone properties of the pixel independent of pixel content and identity; and

(c) determining a set of modified pixel color values for the set of original color values, wherein the set of modified pixel color values equals a non-linear combination of the set of original color values and the set of tone-mask values.

13. The method of claim 12 further comprising:

(a) receiving a set of original color values for each pixel of the input image;

(b) generating a set of tone-mask values for each set of original color values; and

(c) determining a set of modified color values for each set of original color values, wherein each set of modified

color values equals the non-linear combination of a set of original color values and its corresponding set of mask values.

14. The method of claim 13 wherein determining a set of modified color values includes using a set of original color values and its corresponding set of mask values as indices into a look-up table to retrieve the set of modified color values.

15. The method of claim 13 wherein determining the set of modified color values includes using a non-linear operation to combine each value in the set of original color values with its corresponding mask value.

16. The method of claim 15 wherein the non-linear operation is an exponential operation having a base and an exponent, said operation using a value from the set of original color values as part of its base, and a value from the corresponding set of mask values as part of its exponent.

17. The method of claim 13 wherein generating the tone-mask values includes using a non-iterative process.

18. The method of claim 13 wherein generating the tone-mask values includes:

generating sets of monochrome color values from the sets of original color values,

inverting the sets of monochrome color values, and

performing a low-pass filtering operation on the sets of inverted, monochrome color values.

19. A computer readable medium having a set of instructions stored therein for enabling a computer to perform local color correction on an input image having a plurality of pixels, said set of instructions including:

(a) a first set of instructions, which when executed by the computer cause the computer to retrieve a set of original color values for each pixel of the input image,

(b) a second set of instructions, which when executed by the computer cause the computer to generate a set of tone-mask values for each set of original color values, and

(c) a third set of instructions, which when executed by the computer cause the computer to determine a set of modified color values for each set of original color values, wherein each set of modified color values equals a non-linear combination of a set of original color values and its corresponding set of mask values.

20. The computer-readable medium of claim 19 wherein the third set of instructions cause the computer to use a set of original color values and its corresponding set of mask values as indices into a look-up table to retrieve a set of modified color values.

21. The computer-readable medium of claim 19 wherein the third set of instructions cause the computer to use a non-linear operation to combine a set of original color values and its corresponding set of mask values.

22. The computer-readable medium of claim 21 wherein the non-linear operation is an exponential operation having a base and an exponent, said operation using the set of original color values as part of its base, and the corresponding set of mask values as part of its exponent.

23. The computer-readable medium of claim 19 wherein the second set of instructions generates the tone-mask values by using a non-iterative process.

24. The computer-readable medium of claim 19 wherein the second set of instructions includes:

a fourth set of instructions that cause the computer to generate sets of monochrome color values from the sets of original color values,

a fifth set of instructions that cause the computer to invert the sets of monochrome color values, and

a sixth set of instructions that cause the computer to perform a low-pass filtering operation on the sets of inverted, monochrome color values.

25. The computer-readable medium of claim 19 wherein each set of original color values is a set of component color values.

26. The computer-readable medium of claim 19 wherein each set of mask values is a grayscale mask value.

* * * * *

CERTIFICATE OF SERVICE

I, the undersigned, hereby certify that on May 23, 2008, I electronically filed the foregoing with the Clerk of the Court using CM/ECF, which will send notification of such filing(s) to the following:

William J. Marsden, Jr.
FISH & RICHARDSON P.C.

I also certify that copies were caused to be served on May 23, 2008 upon the following in the manner indicated:

BY E-MAIL

William J. Marsden, Jr.
FISH & RICHARDSON P.C.
919 N. Market Street, Suite 1100
Wilmington, DE 19801

Matthew Bernstein
John E. Giust
MINTZ LEVIN COHN FERRIS GLOVSKY AND POPEO PC
5355 Mira Sorrento Place
Suite 600
San Diego, CA 92121-3039

Bradley Coburn
FISH & RICHARDSON P.C.
One Congress Plaza, Suite 810
111 Congress Avenue
Austin, TX 78701

Daniel Winston
CHOATE HALL & STEWART, LLP
Two International Place
Boston, MA 02110

/s/ Julia Heaney (#3052)

Julia Heaney (#3052)